Community Costs in Neighborhood Help Problems

Sarah Kühn[a,*], Nadja Stroh-Maraun[a]

[a] *Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany*

Abstract

We define neighborhood help problems where agents may seek and/or provide various kinds of help as one-sided matching markets with incompatibilities. To obtain a Pareto efficient outcome the top trading cycles mechanism (TTC) (Shapley and Scarf, 1974) may be used. However, a short supply of compatible helpers may result in many agents being unmatched forcing them to rely on costly outside options. These agents leave the market without helping and a lot of potential is lost. To overcome this issue we introduce the so-called pool option. This pool gives agents an incentive to provide help when being helped outside of the market. We propose the neighborhood top trading cycles and chains mechanism that incorporates the pool option and is based on the TTCC by Roth et al. (2004). The mechanism is Pareto efficient and strategy-proof. Additionally, it (weakly) reduces overall costs compared to the TTC.

Keywords: Matching, School Choice, Kidney Exchange, Top Trading Cycles, Pareto Efficiency, Strategy-Proofness
JEL: C78, D47

*Corresponding author.

**This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre 901 "On-The-Fly Computing" (SFB 901) under the project number 160364472-SFB901.

Email addresses: `Sarah.Kuehn@uni-paderborn.de` (Sarah Kühn), `Nadja.Stroh-Maraun@uni-paderborn.de` (Nadja Stroh-Maraun)

Preprint submitted to 33rd Stony Brook International Conference on Game Theory 2022 *April 13, 2022*

1. Introduction

A unprecedented wave of solidarity with vulnerable groups was observed in 2020 COVID-19 pandemic despite the severe negative change in all our lives worldwide. In a lot of communities around the globe younger and healthy people helped their vulnerable neighbors with the grocery shopping, fought loneliness by talking to each other on the balcony, looked after their neighbors' children and helped with home schooling. But soon a problem arouse. How can we coordinate these activities and bring people in need together with people willing to help. One idea that was used a lot was the usage of social media groups that were created[1] but the problem was only solved partly. Especially the elderly were not able to use these groups, younger people had the problem of choosing the correct platform and the volunteers that hosted these platforms put a lot of work and effort in matching the people by hand.

Another idea is to consider the neighborhood help as a matching market. On this market we have agents who seek help, agents who offer help and agents who offer to help but need some other type of help themselves. It also matters whom they are matched with. As the type of help needed and offered differs across the market, it is not sufficient that a neighbor wants to help, she also needs to be able to help in a particular way. In a matching market this would mean that some agents are unacceptable or incompatible to other agents. How can we find a Pareto efficient matching in such a neighborhood help market with incompatibilities?

In this paper we introduce the neighborhood help problem and discuss two different help exchange mechanisms that find a Pareto efficient outcome based on the characteristics of the market. More specific, these two markets differ in the incorporation of a so-called pool option. In a first step we show that the top trading cycles algorithm which was introduced by Shapley and Scarf (1974) and adopted by Abdulkadiroğlu and Sönmez (2003) might be used to solve the problem but at potentially high costs if there are a lot of incompatibilities. In this case, each agent in need has only few acceptable matches and a high risk of finding no match. In this case, unmatched agents need to make use of a potentially costly outside option. Imagine the following small example: if parents are looking for a neighbor who helps the children with home schooling and offer help with the grocery shopping in exchange, this may lead to many unacceptable matches as tutoring is not easy. If the parents do not find a match in the neighborhood they will leave the market without helping someone else but now in need to find a private tutor. A different approach could be to incorporate this outside option into the market and reduce costs by doing so. In this example the parents would stay in the market and help someone else with the grocery by using the pool option. This pool could be a community fund that finds and pay the private tutor.

We introduce the neighborhood top trading cycles and chains mechanism (NTTCC) which is based on the work by Roth et al. (2004) and incorporates the pool option. We show that this actually reduces the overall costs and thus, increase social welfare.

The market we study shares some characteristics with the kidney or organ exchange models (Roth et al., 2004, 2005; Biró et al., 2009; Ergin et al., 2017). Pairs of patient and incompatible donor are in need of a compatible donor and offer an organ in exchange. Thus, these models incorporate a similar structure of pairs as agents might seek help (as patient) and provide it (as donor). Additionally, in organ exchanges as well as neighborhood help preference relations are based on compatibilities and incompatibilities such as blood types or the ability to help a certain seeker. The first matching mechanism introduced for kidney exchange is the top trading cycles and chains mechanism (Roth et al., 2004). It works similar to the top trading cycle but introduce the waiting list which enables a prioritization for a cadaver kidney in exchange for the paired incompatible living donor. While this increases the utilitization rate of living donor kidneys it only captures two kinds of agents, patient-donor pairs and donors of cadaver kidneys. We extent these considerations and allow for three types of agents, namely helpers, seekers and seeker helpers. Our NTTCC mechanism generalizes the algorithm by Roth et al. (2004) to allow for the three types of agents. It is always strategy-proof and yields a Pareto efficient outcome. Thus, our model is more general and incorporates the kidney exchange market as a special case.

[1] As an example have a look at https://www.hoch-stift.de.

As the organ exchange, the considered neighborhood help market exhibits many characteristics of classic matching markets (Gale and Shapley, 1962; Shapley and Scarf, 1974). While, in contrast to two-sided markets (Gale and Shapley, 1962), one-sided markets, as. e.g. the housing market (Shapley and Scarf, 1974), suppose that only one side of the market has preferences, school choice models use some kind of hybrid structure and allocate students with preferences to schools with priorities (Abdulkadiroğlu and Sönmez, 2003; Abdulkadiroğlu et al., 2005; Erdil and Ergin, 2008; Chen et al., 2016; Hoyer and Stroh-Maraun, 2020). Our model borrows this concept as helpers prioritize seekers although it is basically a one-sided matching market. Thus, it is straightforward to use algorithms based on top trading cylces like introduced by Abdulkadiroğlu and Sönmez (2003) especially as the TTC is more robust regarding changes (Stroh-Maraun, 2020).

Our study also relates to the idea of time banks to coordinate community help (Seyfang, 2003; Andersson et al., 2021). Time exchanges can be modelles as a many-to-many matching markets with upper quotas regulating it (Andersson et al., 2021). The authors provide an exchange maximizing mechanism which selects an individual rational, efficient and balanced allocation. They classify goods which can be acceptable or unacceptable for each agent. Our approach, on the one hand, is more restrictive, as we consider a one-to-one market. On the other hand, it also is more general as agents may not need to offer help in return to be helped by the community.

2. Model

We consider a neighborhood help problem where neighbors may seek help from another neighbor and also offer some help. Formally, let $A = \{1, ..., n\}$ be the finite set of agents or neighbors. Each agent $i \in A$ either seeks help or offers help or both. Without loss of generality agents are ordered in the following way $A = \{1, ..., k, k+1, ..., m, m+1, ..., n\}$. We can differentiate three types of agents: helper seekers ($\{1, ..., k\} \in A$) offer a help and seek a help, only seekers ($\{k+1, ..., m\} \in A$) only seek help, and altruistic helpers ($\{m+1, ..., n\} \in A$) only offer a help.

We may rewrite all helper seekers i with $i \leq k$ as a tuple (s_i, h_i). If an agent $i \in A$ is a helper, we denote this by h_i. If an agent $j \in A$ is a seeker, we denote this by s_j. Additionally, we introduce two dummies: s_0, if an agent is no seeker, and h_0, if an agent is no helper. Thus, an only seeker j with $k+1 \leq j \leq m$ is denoted as a tuple (s_j, h_0) and an altruistic helper l with $m+1 \leq l \leq n$ is denoted as a tuple (s_0, h_l). Now we can transform the one set of agents into two disjoint sets of helpers and seekers. The set of seekers $S = \{s_1, ..., s_k, s_{k+1}, ..., s_m\}$ includes the seeking part of helper seekers ($\{s_1, ..., s_k\} \in S$) and the only seekers ($\{s_{k+1}, ..., s_m\} \in S$). The set of helpers $H = \{h_1, ..., h_k, h_{m+1}, ...h_n\}$ includes all helpers who either are also a seeker ($\{h_1, ..., h_k\} \in H$) or who help altruistically ($\{h_{m+1}, ...h_n\} \in H$).

We assume that there is an outside option, the pool p which offers the opportunity to get help outside the neighborhood help market. We assume that being matched to the pool incurs costs of c per seeker in the pool.

Each seeker $s_i \in S$ has complete and transitive preferences R_i over all helpers and the pool, $H \cup p$, so that they can be represented by ordered lists. For each seeker s_i the set of helpers H can be partitioned into two disjoint sets containing compatible and incompatible agents. Agents compatible with s_i are denoted by $H_i \subseteq H$. s_i is indifferent between all incompatible helpers, and strictly prefers all compatible agents over either the pool p or h_i. Thus, an only seeker i has strict preferences P_i over $H_i \cup \{p\}$. A helper seeker has strict preferences P_i over $H_i \cup \{p, h_i\}$. h_i may be part of H_i or $H \backslash H_i$. To compare two options we introduce the following notation: denote by $h_a \succ_{s_i} h_b$ that s_i prefers h_a over h_b.

Helpers do not have preferences. Instead we assume that each helper $h_j \in H$ has a complete, irreflexive and transitive binary priority Π_j over all seekers S. Therefore, helpers are not treated as agents in the problem but as objects. We assume that helper seekers give the highest priority to themselves, formally for all h_j with $j \in \{1, ..., k\}$ it holds that $s_j \Pi_j s_i$ for all $s_i \in S$ with $i \neq j$.[2] The market of the neighborhood help problem is now defined as $M = (S, H, P, \pi)$.

[2]We borrow the ideas of priorities from the school choice literature (e.g. Abdulkadiroğlu and Sönmez, 2003) where schools are

An outcome of the neighborhood help problem assigns each seeker to either one helper or the pool and each helper to at most one seeker. If a seeker is matched to a helper, then this helper is also matched to this seeker and vice versa. Please notice that the pool option can be assigned multiple times.

Definition 1. *More formally, a matching is a mapping* $\mu : S \cup H \to S \cup H \cup \{p\}$ *such that*
$\mu(s) \in H \cup \{p, s\}$ *for all* $s \in S$,
$\mu(h) \in S \cup \{h\}$ *for all* $h \in H$, *and*
$\mu(s) = h$ *if and only if* $\mu(h) = s$.

A *help exchange mechanism* selects a matching for each neighborhood help problem. As we have seen before, seekers are interpreted as agents, helpers are not. This has several consequences. First of all, to evaluate a matching, the interests of the seekers are decisive. Second, helpers do not act strategic. In the following we will define properties of the matching and the mechanism that take these considerations into account.

Each seeker i may compare their matches in two different matchings, μ and ν. They weakly prefer matching μ over matching ν, denoted $\mu(i) \succsim_i \nu(i)$, if and only if they weakly prefer the match in μ over the match in ν, formally $\mu(i) R_i \nu(i)$ or $\nu(i) = \mu(i)$.

Definition 2 (Pareto efficiency (Abdulkadiroğlu et al., 2017))**.** *A matching* μ *Pareto dominates a matching* ν *if* $\mu(s) \succsim_s \nu(s)$ *for all* $s \in S$ *and* $\mu(s) \succ_s \nu(s)$ *for some* $s \in S$*. A matching is Pareto efficient if it is not Pareto dominated by any other matching.*

A matching is Pareto efficient if it is not possible to improve a seeker's allocation without worsening another seeker's situation. Note that helpers are not taken into account in Pareto efficiency considerations.

To discuss strategic behavior, we define the neighborhood help problem as a preference revelation game induced by a particular help exchange mechanism φ. The matching selected by φ is denoted as $\varphi(P)$ given the preferences P. $\varphi_s(P)$ denotes the match of seeker$s$ in $\varphi(P)$.

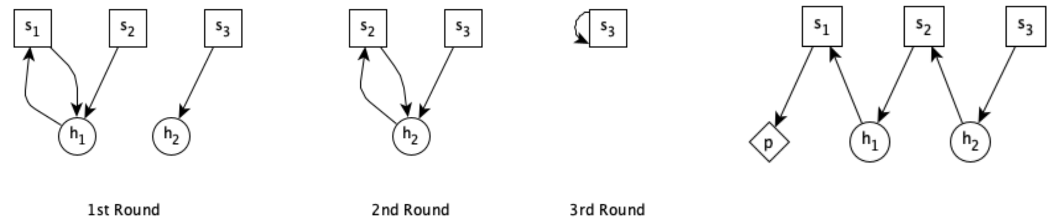
Definition 3 (Strategy-proofness (Abdulkadiroğlu et al., 2017))**.** *A mechanism* φ *is strategy-proof if it is a dominant strategy for each seeker* s *to state the true preferences* P_s *in the preference revelation game induced by* φ*. Formally, for all* P *and* s *and all possible misrepresentations* P'_s *of the preferences it holds that* $\varphi_s(P) \succsim_s \varphi_s(P'_s, P_{-s})$.

If a mechanism is strategy-proof, all seekers in the neighborhood help problem are able to state their true preferences without being harmed.

3. Help Exchange Mechanisms

Help exchange mechanisms select a matching for each neighborhood help problem. In the following section we discuss two different strategy-proof help exchange mechanisms that both find Pareto efficient matchings. Nevertheless, they differ in the incorporation of the pool option p. First, we shortly discuss the well-known top trading cycles algorithm (Shapley and Scarf, 1974) which does not incorporate the pool option p. Afterwards, we introduce the neighborhood top trading cycles and chains mechanism (NTTCC) which is based on the top trading cycles and chains algorithm by Roth et al. (2004) and gives seekers the opportunity to be matched to the pool option p. The pool is a special kind of outside options. In classical matching models, outside options are simply alternative options for agents and come into play if these agents stay unmatched in a matching. In a neighborhood help problem the outside option might be either staying unmatched, being matched to oneself or being matched to the pool p. We assume that a seeker who gets one of this three outside options needs help from outside the neighborhood. For simplicity, this help always incurs fixed costs c per seeker. In this setting, the pool might then be interpreted as a community attempt to help all neighbors that are not matched to another neighbor. We show later in the section that it is possible to reduce the overall costs of a neighborhood help problem by using the pool option.

not treated as agents but as objects with some priority ordering derived from some objective criteria, such as walking distance from school or the fact whether a student already has siblings studying at the school. Here, we assume that helpers are treated as objects as their only interest is to help the other neighbors. Thus, the priority ordering might be based on some criteria like the proximity of a seeker's home. We will later see that this priority ordering might also be a common list based on the seekers' needs and neediness.



(a) Outcome of the TTC

(b) Alternative allocation

Figure 1: Two possible matchings

3.1. Top Trading Cycles Algorithm in the Neighborhood Help Problem

Let us start with having a look at the TTC again. It was introduced by Shapley and Scarf (1974) in a setting where a group of agents, each of them equipped with a house, wants to exchange their houses within the group. Later the algorithm was adopted to fit other models like the school choice problem (Abdulkadiroğlu and Sönmez, 2003) to allocate students to schools based on their preferences and the schools' priorities. In each round every student is pointing at their most preferred school among all available schools and every school is pointing at the student that is mostly prioritized among all pointing students. If there is a cycle, each student is matched to the school they are pointing at. The algorithm always yields a Pareto efficient outcome and it is strategy-proof for the students (Abdulkadiroğlu and Sönmez, 2003).

It is easy to see that the TTC is also applicable to the neighborhood help problem if the pool option is not implemented directly. This means, in each round every seeker is pointing to a helper according to their preferences and every helper is pointing to a seeker according to their priorities. If seekers prefer the pool option over all helpers in the mechanism, they are pointing at either their own helper (if it is a seeker helper) or towards themselves. In either way, a cycle is formed and the seeker is not matched to someone else, leaves the market and may use the outside option. Nevertheless, there is one problem that we can identify. If a seeker prefers the outside option, this seeker leaves the market as well as the corresponding helper in case of a seeker helper pair. It seems like a lot of potential is lost here as this seeker's helper will not help. To illustrate the problem, imagine the following small Example 1.

Example 1. *There are three seekers left in the mechanism, seeker s_1 brings helper h_1, seeker s_2 brings helper h_2, and the last one, seeker s_3, is an only seeker. s_1 prefers the pool option, s_2 prefers h_1 over the pool, s_3 prefers h_2. According to the TTC, s_1 is matched to the pool and leaves the market together with h_1. s_2 and s_3 are then matched to the outside option as well. As we assume that the pool option is costly, total costs of 3c arise. If we assume that each seeker has to pay individually for the pool option, the costs every seeker has to pay are given by c. This seems odd as h_1 leaves the market without helping anyone although s_2 wanted their help. The same is true for h_2 and s_3. So, another possible matching would be here that s_3 is matched to h_2, s_2 is matched to h_1 and s_1 is matched to the pool, reducing the overall costs for the outside option to c. To incentivize h_1 to stay in the mechanism although s_1 gets the outside option, we could think of dividing the overall costs among all of the seekers. In this case, everyone is better off than in the TTC matching. Both allocations are depicted in Figure 1.*

In the next section we want to show a way to actually allow for this kind of exchanges with the help of the NTTCC.

3.2. Neighborhood Top Trading Cycles and Chains Mechanism

If there exists a lot of incompatibilities, the number of acceptable matches for each seeker may be small. Thus, the TTC might find few cycles incorporating more than one original agent $a \in A$. As a consequence many seekers will not get help from others but stay unmatched (or more precisely are matched to themselves). To overcome this issue and match more seekers to helpers we have a closer look at the pool. Please remember, the pool option p offers another outside option. We can imagine the pool as a neighborhood project. Everyone, who is matched to the pool after the neighborhood help mechanism was used, gets help from an outside option. Until now in the TTC we could guarantee that there is always a cycle as every seeker was forced to always point to a helper or themselves. With the introduction of the pool this is no longer the case as the pool does not point to anyone. If a seeker points to the pool, this seeker therefore cannot be part of a cycle. Instead, the seeker is part of a *chain*.

Definition 4. *A chain consists of a number of seekers, a number of helpers and the pool option p such that each helper is pointing to a seeker and each seeker except for one is pointing to a helper. The other seeker, who is not pointing to a helper, points to the pool. A chain always starts with a helper or an only seeker, we call them first agent in the chain, and ends with the pool.*

Please notice that each seeker, each helper and the pool might be a part of more than one chain at the same time. The idea of chains was introduced by Roth et al. (2004) in the context of kidney exchanges where patients in the model had the possibility to enter the waiting list for cadaver kidneys instead of being matched to an incompatible donor. They have shown that in a directed graph where patients and donors as well as the waiting list are nodes and every patient points to a donor or the waiting list and every donor points to its paired patient, there either exits a cycle or a chain (Roth et al., 2004, Lemma 1). It is easy to see that this also holds in our model when each seeker either points to a helper or the pool and each helper points to a seeker.

After we have defined chains, we are now able to introduce the neighborhood top trading cycles and chains (NTTCC) mechanism that is based on the top trading cycles and chains algorithm by Roth et al. (2004) and incorporates the pool option p. It works as follows.

NTTCC

Initially, all seekers and helpers are taking part in the market. Additionally, all seekers and helpers are active and available.

Step 1. As long as there are active seekers remaining in the market, all these seekers s point to their most preferred available helper h or the pool p, if this is preferred over all available helpers. All active helpers h point to the most prioritized available seekers s. There is either a cycle or a chain or both.

Step 2. If there is a cycle, all seekers within the cycle are matched to the helper they point to. Remove all now matched seekers and helpers from the market. Continue with Step 1.

Step 3. If there is no cycle, check whether there is a chain. Order the seekers into a tie-breaking list. This can be done randomly or be based on other criteria. If there are one or more chains, select the chain which starts with the first seeker from the tie-breaking list. The assignment is final for each seeker in the selected chain. However, no seeker or helper is removed from the market to allow the chain's extension. Instead, all agents in the pool and the selected chain are no longer active and all but the first agent of the selected chain are not available. Continue with Step 1.

Step 4. If there is neither a chain nor a cycle, this means that there are no active seekers. The mechanism terminates. All seekers are matched to either a helper or the pool, all still active helpers stay unmatched.

The main properties of the TTC carry over, Pareto efficiency and strategy-proofness.

Theorem 1. *The NTTCC algorithm yields a Pareto efficient outcome.*

Proof. Consider the neighborhood top trading cycles and chains algorithm. Any seeker who leaves the market in the first round is matched to their top choice and thus cannot be made better off. In a proceeding round, any seeker who leaves the market is assigned to their top choice among all available helpers. As their preferences are strict, the seeker cannot be better off without hurting another seeker who left the market (and thus was matched) in a prior round. □

The allowance of only seekers, altruistic helpers, and the pool option, or more precisely, the introduction of chains, does not affect the Pareto efficiency here. Thus, the proof is similar to the one by Abdulkadiroğlu and Sönmez (2003) for the TTC or the one by Roth et al. (2004) for the TTCC.

Additionally, the NTTCC is also strategy-proof. To show strategy-proofness, we use the following Lemma 1.

Lemma 1. *Fix the stated preferences of all seekers except s_i at P_{-i}. Suppose that in the NTTCC algorithm, seeker s_i leaves at round k under P_i and at round k' under some P'_i. W.l.o.g. $k \leq k'$. Then the remaining active seekers and available helpers at the beginning of round k are the same whether seeker s_i announces P_i or P'_i.*

Proof. Seeker s_i is not part of a cycle or a chain in both cases prior to round k. As all the other seekers do not change their preferences, the same cycles and chains are formed prior to round k in both cases. According to the tie-breaking list, the same chains are also selected prior to round k. Thus, the same seekers are active and the same helpers are available in round k, independent of whether seeker s_i states P_i or P'_i. □

With the help of Lemma 1 we can now formulate Theorem 2.

Theorem 2. *The NTTCC algorithm is strategy-proof.*

Proof. Suppose that seeker s_i has true preferences P_i. We fix the stated preferences of all other seekers P_{-i}. If s_i states their true preferences P_i, the seeker is matched in round k. If s_i states some other preferences P'_i, the seeker is matched in round k'. Now we show that misrepresenting their preferences does not make the seeker better off compared to stating the true ones. We have to consider two different cases.

Case 1: $k < k'$

According to Lemma 1 we know that at round k the same seekers are active and the same helpers are available in the market independent of s_i's stated preferences. If s_i states their true preferences, they are matched in k to their most preferred available helper $\mu(i)$. Otherwise, they would be matched later and, thus, would be matched to a less preferred helper, $\mu^*(i) \succsim \mu(i)$.

Case 2: $k \geq k'$

There are two possibilities how a seeker s_i is assigned and subsequently is no longer active. s_i can either join a cycle or a chain. These two cases are considered separately. In both cases we can show that $k = k^{'}$ and by that is weakly better off by pointing to their most preferred helper.

Case 2a:

Under $P^{'}_i$, seeker s_i is assigned by joining a cycle and subsequently is inactive. W.l.o.g. cycles may include any kind of helper (seeker helper or altruistic helper).

Suppose the following cycle: helper h_1 is pointing to seeker s_2, s_2 is pointing to helper h_3 and so on until helper h_r is pointing to seeker s_i and s_i is pointing to h_1. Therefore, seeker s_i would be assigned to h_1 under $P^{'}_i$. We now consider round $k^{'}$. Again, according to Lemma 1 we know that the same seekers are active and the same helpers are available whether s_i announces $P^{'}_i$ or P_i. Thus, at round $k^{'}$ we get the considered cycle that includes s_i. As the preferences of all helpers and seekers in this cycle beside s_i stay unchanged, they will keep on pointing as stated until s_i is no longer available. Considering s_i truthfully points to their most preferred helper, they either receive a helper better than h_1 or is matched to h_1.

Case 2b:

Under $P^{'}_i$ seeker s_i is assigned by joining a chain in round $k^{'}$ and is inactive in the following rounds. Suppose a chain will be selected by a priority ordering under $P^{'}_i$ that includes seeker s_i in round $k^{'}$. We know that seeker s_i is included in this chain if they are the highest prioritized seeker or the chain will start with the highest prioritized seeker and passes through seeker s_i before ending with p. Thus, s_i receives help by a helper h_j or the pool p.

Suppose now that seeker s_i states their true preferences P_i and we are in round $k^{'}$. By Lemma 1 we know that the same patients remain active and the same helpers remain available, whether seeker s_i announces P_i or $P^{'}_i$. Regardless of the stated preferences of seeker s_i, whether they announce their true preferences P_i

or any other preference $P_i^{'}$, seeker s_i will receive a final allocation in round $k^{'}$. As every cycle and every chain with higher priority would have been matched in a round prior to $k^{'}$, there will be no cycle (Case 2b) or one cycle (Case 2a) including seeker s_i. We know that in each round there either exists a cycle or chain if there is at least one active seeker left. By $P_i^{'}$, round $k^{'}$ is the last round of s_i and s_i is assigned to h_j or p via a chain. By P_i, round $k^{'}$ is also the last round where s_i is either assigned via a cycle or a chain. If there is a cycle, the assignment is final in round $k^{'}$ and seeker s_i is matched. If there are one or more chains, one is selected via the priority list. We know that the chain including the highest prioritized seeker passes through seeker s_i. Otherwise the chain would have been selected in a prior round. By the selected chain the assignment of seeker s_i is finalized in round $k^{'}$. Thus, by P_i seeker s_i receives an assignment that is at least as good as h_j or is assigned to h_j under $P_i^{'}$. □

Again, the original TTC and the TTCC seem to be quite robust to the extensions we made. Thus, the proof is again similar to the one by Abdulkadiroğlu and Sönmez (2003) for the TTC and to the one by Roth et al. (2004) for the TTCC.

4. Costs

Applying the NTTCC in contrast to the TTC offers the seekers an additional outside option. By that a helper seeker might offer his help in exchange for receiving help by the pool while they would leave the market unmatched in the TTC. Therefore, we identify in the following which differences between the TTC and the NTTCC enable a weakly reduction in the overall cost. To do so, we first show that there is a close connection between the two mechanisms. The TTC is basically a special case of the NTTCC where only short chains are considered. In the NTTCC, a chain is selected based on the tie-breaking list of the seekers and can be extended in the following rounds. The TTC is equivalent to a version of the NTTCC where the minimal chain is chosen and is removed immediately. Roth et al. (2004) show that the TTCC mechanism which chooses and immediately removes the minimal chain, is strategy-proof (Roth et al., 2004, Theorem 1). They further show that the outcome of the TTC is the same as the outcome of the TTCC with minimal chain selecting and removal (Roth et al., 2004, p.486). By that we show the following:

Proposition 1. *The TTC is equivalent to a version of the NTTCC where the minimal chain is chosen and is removed immediately.*

Proof. Consider the NTTCC, suppose in Step 3 the minimal chain is selected and removed immediately and we do not use a tie-breaking list and do not keep the first agent active. By that each chain consists of either an only seeker who is pointing to the pool p or a helper seeker where the helper h_i points to their seeker s_i who points to the pool p. This is equivalent to a seeker pointing to themselves or their own helper in the TTC. In both cases, the seekers do not directly get help by a neighbor. Costs of c occur for each selected minimal chain.

The only difference that might occur during the application of the mechanisms is the round in which the matching of a seeker pointing to themselves, the own helper or the pool is finalized. In the TTC each agent pointing to their own house or to themselves will be removed directly in round l. In the NTTCC with minimal chain selection and removal, each agent pointing to the pool p would be matched after all other cycles are finalized. Suppose this is in round $l^{'}$. Therefore, $l \leq l^{'}$ as in round l the seeker starts pointing to their own helper or themselves in both mechanisms and is removed in that round (TTC) or points to p and keeps on pointing in the same way (NTTCC with minimal chain selection and removal) until all other cycles are matched and removed. Then the minimal chains are selected and removed next (round $l^{'}$). This does not affect or change the matching as each agent pointing to their helper, themselves or p will keep pointing until the match is selected and removed. Therefore, the TTC and the NTTCC with minimal chain selection and removal yield the same matching. □

Now we are able to compare the two mechanisms, TTC and NTTCC, and the resulting costs c for each seeker who gets help by an outside option. Therefore, we compare the resulting matchings of the two mechanisms and compare the incurred costs.

Theorem 3. *The NTTCC algorithm compared to the TTC weakly reduces the overall costs $\sum c$.*

Proof. Assume that a seeker selecting p yields the same costs c as a seeker preferring to be unmatched or matched to themselves. Considering the TTC, we know that c equals the number of unmatched agents and agents matched to themselves. For both mechanisms, the last agent before the pool p who is seeker s_i induces a cost of c as they need an outside help whether they are unmatched, matched to themselves or matched to the pool p. For the NTTCC, it is easy to see that each time a chain is extended in a later round or includes more than one seeker, the cost c might reduce in comparison to the TTC (NTTCC with minimal chain selection and removal) as there might be a seeker in this chain who would need an outside option otherwise.
A chain can only be extended if a helper points to the last seeker s_i. If no helper points to s_i, the c cannot be reduced by using the NTTCC. If a helper h_i of a helper seeker points to their seeker s_i who points to p and the chain is extended in the following, an additional helper is available in comparison to the removal of the minimal chain from h_i to s_i to p. Therefore, we show that an additional helper weakly reduced the number of agents inducing costs by needing an outside help. □

5. Conclusion

We have shown that both the TTC as well as the NTTCC are reliable mechanisms to match neighbors on a neighborhood help market. However, the two mechanisms differ in an essential point: the incorporation of a pool. Without the pool unmatched helper seekers who leave the market do not only face costly outside options but leave the market without helping other seekers on the market. The NTTCC algorithm offers such agents the pool option to reduce the agent's costs and increase the supply of help.

Additionally, the algorithm is able to (weakly) reduce community costs when matching helpers and seekers on such markets.

References

Abdulkadiroğlu, A., Che, Y.-K., Pathak, P. A., Roth, A. E., and Tercieux, O. (2017). Minimizing justified envy in school choice: The design of new orleans' oneapp. Technical report, National Bureau of Economic Research.

Abdulkadiroğlu, A., Pathak, P. A., and Roth, A. E. (2005). The new york city high school match. *The American Economic Review*, 95(2):364–367.

Abdulkadiroğlu, A. and Sönmez, T. (2003). School choice: A mechanism design approach. *The American Economic Review*, 93(3):729–747.

Andersson, T., Cseh, A., Ehlers, L., and Erlanson, A. (2021). Organizing time exchanges: Lessons from matching markets. *American Economic Journal: Microeconomics*, 13(1):338–73.

Biró, P., Manlove, D. F., and Rizzi, R. (2009). Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Mathematics, Algorithms and Applications*, 01(04):499–517.

Chen, Y., Liang, Y., and Sönmez, T. (2016). School choice under complete information: An experimental study. *The Journal of Mechanism and Institution Design*, 1(1):45–82.

Erdil, A. and Ergin, H. (2008). What's the matter with tie-breaking? improving efficiency in school choice. *American Economic Review*, 98(3):669–89.

Ergin, H., Sönmez, T., and Ünver, M. U. (2017). Dual-donor organ exchange. *Econometrica*, 85(5):1645–1671.

Gale, D. and Shapley, L. (1962). College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15.

Hoyer, B. and Stroh-Maraun, N. (2020). Matching strategies of heterogeneous agents under incomplete information in a university clearinghouse. *Games and Economic Behavior*, 121:453–481.

Roth, A. E., Sönmez, T., and Ünver, M. U. (2004). Kidney exchange. *The Quarterly journal of economics*, 119(2):457–488.

Roth, A. E., Sönmez, T., and Ünver, M. U. (2005). Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151–188.

Seyfang, G. (2003). 'with a little help from my friends.'evaluating time banks as a tool for community self-help. *Local Economy*, 18(3):257–264.

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

Stroh-Maraun, N. (2020). Pareto efficiency in weighted school choice problems. Technical report, Paderborn University, CIE Center for International Economics.