Internally Stable Matchings and Exchanges[*]

Yicheng Liu, Pingzhong Tang and Wenyi Fang
Institute of Interdisciplinary Information Sciences,
Tsinghua University, Beijing, China

Abstract

Stability is a central concept in exchange-based mechanism design. It imposes a fundamental requirement that no subset of agents could beneficially deviate from the outcome prescribed by the mechanism. However, deployment of stability in an exchange mechanism presents at least two challenges. First, it reduces social welfare and sometimes prevents the mechanism from producing a solution. Second, it might incur computational cost to clear the mechanism.

In this paper, we propose an alternative notion of stability, coined *internal stability*, under which we analyze the social welfare bounds and computational complexity. Our contributions are as follows: for both pairwise matchings and limited-length exchanges, for both unweighted and weighted graphs, (1) we prove desirable tight social welfare bounds; (2) we analyze the computational complexity for clearing the matchings and exchanges. Extensive experiments on the kidney exchange domain demonstrate that the optimal welfare under internal stability is very close to the unconstrained optimal.

Introduction

Designing desirable matching and exchange mechanisms has been a topic of intensive researches over the past few years. (cf. (Roth 2000; Parkes and Seuken 2014)). An exchange in a graph is a set of disjoint cycles within which barter exchanges are conducted. A matching is a special exchange where all cycles are of length 2. A desirable feature in such preference-based matching mechanisms, is the so-called *stability* (Gale and Shapley 1962). Roughly, an exchange mechanism is stable if no coalition of agents could deviate to form a new exchange, with everyone in the coalition becoming better off. Stability plays a determinative role in matching-based market design. Roth (2000) surveys 17 major matching-based markets, 10 of which are stable and all successfully survive over time, while among the remaining 7 unstable markets, only 2 survive[1]. As another example, in the organ exchange domain, Dickerson et. al. (2013) observes that, among the assignments suggested by UNOS[2], only 7 percent of which finally make it to surgery. One of the major reasons is that agents find better alternatives in and out of the system.

There are two major challenges that prevent stability from being deployed in the design of an exchange mechanism. For one, imposition of stability as a constraint significantly reduces social welfare of the exchange. We will show that, in certain graphs (called *odd cycles*), a stable outcome does not even exist. In other words, optimal welfare with stability constraints is arbitrarily far from the unconstrained optimal. For the other, imposition of stability incurs computational cost. It is known that in weighted graphs, computing maximum pairwise matchings is in P (Galil 1986), while adding stability constraints turns it into NP-HARD (Feder 1992).

In this paper, we consider a weaker notion of stability, coined *internal stability*. Briefly speaking, Internal stability only requires matched agents to be stable. Minor modification as it might seem, internal stability yields, to a certain degree, satisfactory results with respect to the two challenges. In particular, we study this new notion in four commonly studied matching and exchange settings (to be rigorously defined shortly). Our contribution is summarized in Table 1 and the following subsection.

Our contribution

1. In unweighted graph settings, for pairwise matching, we show that the worst case ratio between maximum internally stable matching and maximum matching is $1/3$. We further show that maximum internally stable matching matches equal number of agents to maximum stable matching when maximum stable matchings exist. The computation problem in this setting has previously been settled by (Tan 1990).
2. In weighted graph settings, for pairwise matching, the worst case ratio between maximum internally stable matching and maximum matching is $\frac{2}{n}$, where n is the number of vertices. We also show that computing a maximum internally stable matching is NP-HARD, based on the hardness result for egalitarian stable roommate problem (Feder 1992).

[*]This work was supported by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the Natural Science Foundation of China Grant 61033001, 61361136003, 61303077, and a Tsinghua Initiative Scientific Research Grant.

[1] A table listing the markets can be found in the full version.

[2] An organ exchange system in the US, www.unos.org.

Table 1: Results of 4 different settings. All bounds are compared to the optimal welfare of the pairwise case.

Setting	Complexity	Tight lower bound
1	P, Known	$1/3$
2	NP-HARD	$2/n$
3	NP-HARD	$\frac{\lfloor T/2 \rfloor}{T}, T = 2\lfloor \frac{L+1}{2} \rfloor + 1$
4	NP-HARD	$2/n$

3. In unweighted graph settings, for L-way exchange[3], the worst case ratio is $\frac{\lfloor T/2 \rfloor}{T}$, $T = 2\lfloor \frac{L+1}{2} \rfloor + 1$. In addition, the computation problem is NP-HARD.
4. In weighted graphs, L-way exchange. The worst case ratio is $\frac{2}{n}$. The computation problem is NP-HARD.

Furthermore, we conduct extensive experiments on data sets generated by the US and China population statistics. Even though some computation problems are NP-HARD, we use an integer program formulation and implementation that clears instances at certain practical levels. Our experiments confirm that the social welfare of optimal internally stable exchange is very close to the unconstrained optimal while consistently beats the standard stability. It is worth pointing out that our IP formulations are of independent interest.

Related work

Anshelevich et. al. (2013) study the welfare loss caused by standard stability constraints and conduct experiments of the welfare loss on randomly generated graphs. In a later work, Anshelevich et. al. (2013) show that, in the dynamic settings, stable mechanisms sometimes can yield better social welfare than greedy mechanisms.

In general, the settings studied in this paper is related to *hedonic games* (Banerjee, Konishi, and Sönmez 2001; Bogomolnaia and Jackson 2002; Dimitrov et al. 2006), where agents have preferences over subsets of agents and the goal is to find some partition of all agents. Internal stability was introduced in a specific hedonic game (Dimitrov et al. 2006), where preferences are restricted to be binary in the sense that an agent is either preferred or not preferred. Each agent prefers to be in a partition with more preferred agents. They give an algorithm for finding core elements via internal stability.

As mentioned, most of our main results are investigated under general matching and exchange contexts. It might be helpful, however, to think of the *kidney exchange* as a potential application (Roth, Sönmez, and Ünver 2004; 2005; Abraham, Blum, and Sandholm 2007). In a kidney exchange mechanism, a pair of incompatible patient and donor seek to exchange kidney with another pair (Roth, Sönmez, and Ünver 2005) or among several other pairs (Roth, Sönmez, and Ünver 2004; Abraham, Blum, and Sandholm 2007), the mechanism returns an exchange that maximizes social welfare, subject to the constraint that no pair can obtain a kidney unless they donate one in return. Over the past few years, kidney exchange has been studied in a number of realistic contexts (Awasthi and Sandholm 2009; Ashlagi and Roth 2011; Ashlagi et al. 2010; Dickerson, Procaccia, and Sandholm 2012a; 2012b; 2014).

Preliminaries

An exchange problem can be modelled as a directed graph $G = (V, E)$ with $|V| = n$, where each vertex $v_i \in V$ represents an agent and each arc $v_i v_j \in E$ states that v_i is acceptable to v_j. Each vertex maintains a linear order preference over all its neighbors. The rank of v_j in v_i's preference list is denoted by r_{ij}. Lower ranking donor is more preferred. We use *edge* to denote a 2-way cycle between two vertices.

We distinguish between the following two cases:

- weighted graph, there is a weight w_{ij} on each arc $v_i v_j$. The weight is consistent with the preference ranking, i.e., $w_{ij} > w_{ik} \to r_{ij} < r_{ik}$.
- unweighted graph, each arc has a weight 1.

An exchange M is defined to be a set of arcs that form vertex-disjoint cycles. Our goal, for both weighted and unweighted graphs, is to find a maximum weighted exchange. That is, $\arg\max_M \Sigma_{v_i v_j \in M} w_{ij}$.

Quite often, there can be cycle length constraints for certain specific domains. For example, in kidney exchange, there is no cycle that should be longer than some length L. When $L = 2$, the exchange problem reduced to the roommate problem (Gale and Shapley 1962). When there is no constraint on L, the *top trading cycle mechanism* (Roth, Sönmez, and Ünver 2004) is known to be stable and strategy-proof.

We now define two notions of stability: the standard one (Gale and Shapley 1962) and internal stability.

Definition 1 *Given a graph $G = (V, E)$, an exchange is stable if there does not exist a subset of vertices that wish to deviate from the current assignment to form a new exchange so that everyone in the subset is better off.*

Internally stability is weaker than the standard one in that it only requires stability among the matched agents.

Definition 2 *Given a graph $G = (V, E)$, an exchange is internally stable if there does not exist a subset of currently matched vertices that wish to deviate from the current assignment to form a new exchange, so that everyone in the subset is better off.*

Clearly, stable exchanges do not always exist. In contrast, internally stable exchanges always exist. It is easy to see that any single edge constitutes an internally stable exchange.

Definition 3 *An exchange is k (internally) -stable, if for any deviating subset vertices within size k, the (internally) stability constraint hold.*

Let *L-k exchange* denote an exchange with cycle length no more than L under k stability. So, a 2-2 exchange in fact stands for *stable pairwise matching*, while a 3-2 exchange stands for a standard 3-way exchange with 2-stable constraints. Without mentioning otherwise, the stability notion under consideration is internal stability.

[3] An exchange that can only produce cycles of length $\leq L$.

Theoretical results

Our analysis in this section covers the following four cases:

1. unweighted graph, stable pairwise matching;
2. weighted graph, stable pairwise matching;
3. unweighted graph, L-k exchange;
4. weighted graph, L-k exchange.

We are interested in the following important questions.

- What is the *worst case ratio* between the welfare of maximum internally stable exchange and unconstrained maximum exchange? That is, the welfare loss caused by internal stability[4].
- What is the complexity of computing a maximum weighted exchange, subject to stability constraint? As mentioned, except for case 1, this complexity is known to be NP-hard under standard stability.

In the following, we report our findings with respect to the two questions in these four cases. We conclude that internal stability is a desirable tradeoff that can be potentially deployed in exchange market design.

Pairwise stable matching in unweighted graphs

It is important to note that the stable pairwise matching problem in unweighted graph is equivalent to the so-called *stable roommate problem* (Gale and Shapley 1962). The stable roommate problem is defined as follows. There are n people, each of which has a preference over acceptable others. The goal is to find a stable assignment with $\frac{n}{2}$ disjoint pairs. As for our problem, the goal is finding a stable assignment containing as many pairs as possible. In stable roommate problem, computing a perfect stable matching with strict preference is shown to be solvable in $O(n^2)$ time (Irving 1985). Replacing stability by internal stability does not change this fact (Tan 1990) but has desirable welfare bound shown in Theorem 2.

The standard stable matching does not exist in an *odd cycle*. Here, an odd cycle is a graph $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ with $n \geq 3$ being odd and $v_iv_{i+1}, v_iv_{i-1} \in E, \forall i = 1, 2, \ldots, n$ with $v_0 = v_n$ and $v_{n+1} = v_1$. No other arc exists in the graph. $r_{i,i+1} = 1, r_{i,i-1} = 2$. It is easy to see that no stable matching exists in such a graph, since there are odd number of vertices in the graph, at least one vertex is left unmatched. Without loss of generality, we denote this vertex by v_i. As $r_{i-1,i} = 1$, v_{i-1}'s best choice is v_i. Both v_i and v_{i-1} get better off by matching with each other. This contradicts the definition of standard stability.

The theorem below shows internal stability is desirable in the context of odd cycles as well as in general graphs. It also serves as the basis for proving the welfare bound in Theorem 2.

Theorem 1 *In any unweighted graph $G = (V, E)$, let M^* denote a maximum internally stable matching and M' denote a maximum stable matching (if exists). We have:*

1. *M^* always exists,*
2. *when M' exists, $|M^*| = |M'|$,*
3. *if G is an odd cycle, M^* has the same size as a maximum matching.*

The proof of Theorem 1 is rather involved. We need some existing results from Irving's Algorithm (1985) and Tan's Algorithm (1990). For completeness, we include both algorithms in the attachment. An overview of Tan's algorithm, which computes maximum 2-2 exchange for unweighted graphs, is as follows:[5]

A vertex b in a's preference list is defined as an entry $(a|b)$. Tan's algorithm consists of 2 phases.

In the first phase, each vertex i proposes to his most preferred vertex j, namely $r_{ij} = 1$. j removes all $(j|k)$ satisfying $r_{jk} > r_{ji}$, and symmetrically all $(k|j)$ are removed. Repeat this procedure until no entry can be removed anymore. Label the vertices whose preference list contains less than 2 elements as inactive. Phase 1 ends here.

In the second phase, define a *rotation* to be two sequences of vertices $(a_1a_2 \ldots a_r|b_1b_2 \ldots b_r)$ satisfying $r_{a_ib_i} = 1$ and $r_{a_{i+1}b_i} = 2$ (subscripts mod. r). Repeat the following 3 steps until no active agent left: (1) Find a rotation $(a_1a_2 \ldots a_r|b_1b_2 \ldots b_r)$. (2) $\forall i = 1, 2, \ldots, r$, a_i proposes to b_{i+1}. b_{i+1}, remove all $(b_{i+1}|k)$ with $r_{b_{i+1}k} > r_{b_{i+1}a_i}$, and symmetrically remove all $(k|b_{i+1})$. (3) Label those whose preference list contains less than 2 elements as inactive.

We make use of a lemma from Tan (1990). The "table" in the lemma contains n rows, where the ith row denotes agent i's preference list.

Lemma 1 *(Tan 1990) Let T_2 be a table in phase 2 and let $R = (a_1, a_2, \ldots, a_r)|(b_1, b_2, \ldots, b_r)$ be a rotation exposed in T_2. Suppose that on elimination of R from T_2, some list becomes empty, then*

- *$r = 2m + 1$ for some m and $b_i = a_{i+m}$ for all i,$1 \leq i \leq$ r(subscripts mod. r);*
- *For each i, there are only two entries in a_i's list in T_2, namely $b_i (= a_{i+m})$ and $b_{i+1} (= a_{i+m+1})$;*
- *the list of a_i, but no other list become empty on elimination of R.*

Call rotation R described in Lemma 1 as an *odd rotation* and the eliminatation of such R an *odd elimination*.

If i has single entry in the preference list, say $(i|j)$, then i and j are matched. For an odd rotation $R = (a_1, a_2, \ldots, a_r)|(b_1, b_2, \ldots, b_r), r = 2m + 1$, $\forall i = 1, 2, \ldots, m$, a_i and a_{i+m} are matched. This matching is a maximum internally stable matching.

Lemma 2 *Let A be the set of all vertices that have empty preference lists after phase 1. The vertices in A must be pairwise disconnected.*

Proof: After the first phase, for any two vertices $x, y \in A$. If there is an edge between them, they are in each other's preference list at the beginning. We claim that (1) the entries

[4] Clearly, this ratio is unbounded for the standard stability since there are cases where stable matching is empty or does not exist.

[5] The only difference between Tan's and Irving's algorithm is that Irving's algorithm terminates and reports no complete matching whenever an empty preference list appears.

between them cannot be eliminated unless one of them is the most preferred vertex by someone; (2) if an agent is most preferred by someone, his preference list can not be empty after phase one.

During the phase 1, the only two cases of elimination are: (1) x prefers y the most, those less preferred than x in y's preference list will be eliminated. (2) if $(x|y)$ is eliminated by the first case, $(y|x)$ is also eliminated.

First we prove the correctness of the first claim. If neither x nor y is most preferred by someone, both $(x|y)$ and $(y|x)$ can not be eliminated by case 1. So, $(y|x)$ and $(x|y)$ cannot be eliminated by case 2.

Now we prove the second claim. Without loss of generality, Let $(x|z)$ be the last eliminated entry satisfying that z prefers x most. Then $(x|z)$ cannot be eliminated by case 2, as x is z's most preferred. If it is eliminated by case 1, someone else t prefers x most, t is still in x 's preference list. This contradicts the assumption that $(x|z)$ is the last. ■

We are now ready to prove Theorem 1.

Proof:

1. As M^* is a maximum internally stable matching, it exists if and only if an internally stable matching exists. We know that $M = \emptyset$ is an internally stable matching. So a maximum internally stable matching exists.

2. By (Irving 1985, Corollary 3.2), we have if one or more preference list is empty after the algorithm, then the original problem instance admits no perfect stable matching. Thus, there is no perfect stable matching if one's preference list is empty during the algorithm.

   - Lemma 1 says that once an empty preference list emerges in phase 2, there is an odd rotation. By (Tan 1990, Theorem3.1 and Theorem 3.7), we know that eliminations of Phases 1 and 2 in the Irving's algorithm can retain at least one maximum stable matching. An odd rotation alone is the same as an odd cycle, which indicates there is no stable matching. So when empty preference list appears in phase 2, this item holds.
   - If a's preference list is empty after phase 1, suppose a and b are connected, from Lemma 2, we know b's preference list is not empty after phase 1. By (Tan 1990, Property 3.3), we have, after phase one, if c is the last on b's preference list, b is the first on c's preference list. So, if b and a are matched, c and b can match each other to get better payoff. So a can't be matched to anyone else. As a can't be in any stable matching. removing a from agents will not affect maximum stable matching, neither do maximum internally stable matching constructed by the Tan's algorithm. So if there is a maximum stable matching, after removing all with empty preference lists after phase 1, a perfect stable matching exists. A perfect stable matching must be a maximum internally stable matching, because no matching can match more vertices than perfect stable matching and all matched vertices are stable among themselves.
   - If no empty set emerges during the execution of the algorithm, there is a perfect stable matching, which is also the maximum internally stable matching.

   This completes the proof for the second item.

3. On an odd cycle of n vertices, as n is an odd number, maximum matching can match at most $n-1$ vertices. Following Tan's algorithm, the whole odd cycle is exactly an odd rotation in phase 2. Tan's algorithm matches $n-1$ vertices. Thus, maximum matching matches as many as maximum internally stable matching on odd cycles.

■

The following result shows the worst case ratio of pairwise matching in unweighted graphs.

Theorem 2 *For pairwise matching in an unweighted graph, the worst case ratio between maximum internally stable matching and maximum matching is $\frac{k-1}{2k}$, where k is the size of smallest odd rotation. In particular, when $k = 3$, the fraction is minimized to be $\frac{1}{3}$.*

Proof: **Lower bound.** First, we show that if a maximum stable matching exists in a graph, the ratio between maximum stable matching and maximum matching is $\frac{1}{2}$. For any edge v_iv_j in a maximum matching, if both of them are not matched in the maximum stable matching, v_i and v_j can match together without affecting others. This contradicts to the definition of maximum stable matching. If a stable matching does not exist, according to the proof of Theorem 1, there exists an odd elimination. In an odd cycle of size k, internally stable matching can match at most $k-1$ of them. If all vertices in the "odd cycle" are matched, it will no longer be internally stable. The odd cycle can be in at most k edges in a maximum matching, so the ratio is $\frac{k-1}{2k}$.

For a given graph $G = (V, E)$ and a maximum internal stable matching M, whenever there is an odd rotation in the graph, we can pick out the odd rotation and match $k-1$ nodes out of the k-cycle. Take this part out of the graph, the remainder in M is still an internally stable matching. Repeat this step until there doesn't exist an odd cycle. As shown before, maximum stable matching matches the same amount of agents as maximum internally stable matching when there is no odd cycles. Each time we pick out an odd cycle, the ratio is $\frac{k-1}{2k}$ and in the final graph without odd cycles, the ratio is $\frac{1}{2}$, so the overall ratio is bounded below by $\frac{k-1}{2k}$.

Tightness. We now prove the ratio is tight via the following instance. Given $G = (V, E)$, where $V = \{a_1, a_2, a_3, b_1, b_2, b_3\}$, for $i = 1, 2, 3$, $a_ib_i, b_ia_i \in E$, for $i, j = 1, 2, 3, i \neq j$,$b_ib_j \in E$. $r_{b_1b_2} = r_{b_2b_3} = r_{b_3b_1} = 1$, $r_{b_2b_1} = r_{b_3b_2} = r_{b_1b_3} = 2$, $r_{b_1a_1} = r_{b_2a_2} = r_{b_3a_3} = 3$. In such a graph, we cannot include any two 2-way cycles in an internally stable matching. While matching $\{a_ib_i, b_ia_i | i = 1, 2, 3\}$ contains three 2-way cycles. So the bound is tight.

Figure 2 shows an instance. In this figure heavier arcs denote higher preferences. ■

Pairwise stable matching in weighted graphs

We now investigate the welfare loss in weighted graphs.

Theorem 3 *In weighted graph, the worst case ratio of pairwise matching is $\frac{2}{n}$.*

Proof: **Lower bound.** In weighted graph, the result between maximum internally stable matching and maximum matching can only reach $\frac{2}{n}$ no matter what the maximum cycle

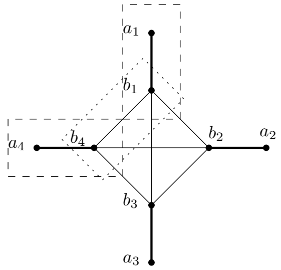


Figure 1: worst case on weighted graph

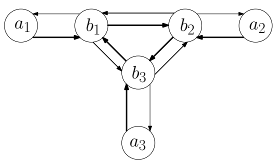


Figure 2: worst case on unweighted graph

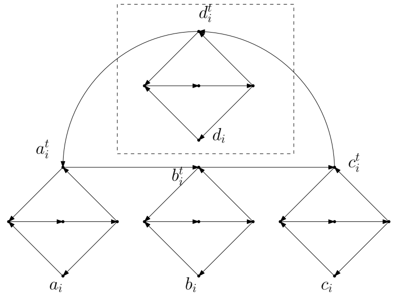


Figure 3: the structure of each 4-vertex super-edge

length is. First, the ratio can be at least $\frac{2}{n}$. Given a graph, we can pick the largest weight edge e only, The matching $\{e\}$ is an internally stable matching. A maximum matching contains at most $\frac{n}{2}$ edges and the weight of e is not less than any edge in the maximum matching.

Tightness. To show this bound is tight. Let w be the weight function. We construct a graph $G = (V, E)$, $V = \{a_1, b_1, a_2, b_2, \ldots, a_m, b_m\}, n = 2m$, $\forall i, a_ib_i \in E$ with $w(a_ib_i)$ near 1. $\forall i, b_ia_i \in E$; $\forall i, j, i \neq j, b_ib_j \in E$; $\forall i, j$, $i \neq j$,$\epsilon > w(b_ib_j) > w(b_ia_i)$, where ϵ is an arbitrarily small number. A matching containing all edges between a_i's and b_i's can reach a social welfare nearly m. A matching containing b_ia_i and $b_ja_j (i \neq j)$ simultaneously cannot be internally stable, as b_i and b_j prefer each other. So the internally stable matching can only contain 1 edge between a_i's and b_i's. This sets the worst case ratio to be $\frac{1}{m} = \frac{2}{n}$.

Figure 1 shows an example where the worst case ratio is exactly $\frac{2}{n}$. Heavier edges have weights near 1[6], the others have weights near 0. The maximum matching obtains all the 4 heavier edges, whereas maximum internally stable matching can obtain at most one. If two heavier edges are in some solution, like the two dashed box in the figure, the two vertices in dotted box get better off by matching with each other. This instance has a worst case ratio of $\frac{1}{4}$. ■

Theorem 4 *Computing the maximum internally stable pairwise matching in weighted graphs is* NP-HARD.

The proof can be found in the full version.

L-k exchange in unweighted graphs

Applying a similar approach as in the previous setting, we obtain a more general tight bound for L-k exchange. The only difference is that for large L, small odd cycles can now be matched directly. So the worst case ratio becomes better but cannot exceed $\frac{1}{2}$.

Theorem 5 *For L-k exchange in unweighted graphs, the worst case ratio is $\frac{\lfloor T/2 \rfloor}{T}$, where $T = 2\lfloor \frac{L+1}{2} \rfloor + 1$, compared to maximum pairwise matching.*

This theorem follows from a similar proof as in Theorem 2. When longer cycles allowed in exchange, we can deal with small odd rotation with a single cycle. The odd rotation begins to take effect when its size is larger than the cycle length limit. T is the minimum size of odd rotation which cannot be matched by a single cycle with length not longer than L.

About the complexity, (Huang 2010) and (Biró and McDermid 2010) show that finding maximum stable 3-way exchange is NP-COMPLETE. Our result allows longer cycle length and requires agents to be stable among any subsets, under length limit.

We prove that, deciding whether L-k exchange matches all agents is NP-COMPLETE. Our proof is inspired by the 3-D MATCHING gadget in (Abraham, Blum, and Sandholm 2007). We reduce from L-D-MATCHING, where L is an arbitrary integer greater than 2. Since 3-D MATCHING is an NP-COMPLETE, so is L-D-MATCHING.

Definition 4 *(Garey and Johnson 1979) L-D-matching refers to the following decision problem: Given L disjoint sets of vertices $X_1, X_2, X_3, \ldots, X_L$ of size q for each, a set of hyper-edged $T \subseteq X_1 \times X_2 \times X_3 \times \ldots \times X_L$, deciding whether there is a disjoint subset S of T of size q, i.e., covering all vertices.*

We show the computation of L-k exchange is NP-COMPLETE by reducing from this problem.

Theorem 6 *For all $L > k \geq 3$, deciding whether all agents can be matched by L-k exchange in unweighted graphs is* NP-COMPLETE.

Proof: First, given a solution, we can easily verify it in polynomial time. It indicates this problem is in NP.

We construct a graph from an instance of L-D-MATCHING as follows. For each super-edge $e_t = e_{t1}, e_{t2}, e_{t3}, \ldots e_{tL}$, we construct a subgraph for each e_{ti}. In each subgraph, there are a sequence of L-1 vertices, which can be used to form an L-way cycle with e_{ti} or e_{ti}^t, just like the part in the dashed box in Figure 3. After that, we line up all e_{ti}^t in the same direction as shown in the Figure 3.

If there is an L-k exchange, we can convert it to a solution for L-D-matching. If $\{e_{ti}^t$ for $1 \leq i \leq L\}$ is an L-cycle in the solution of maximum L-k exchange problem, e_t is in the result of L-D-matching. If the maximum L-k exchange cannot cover all vertices, it implies there is no solution for the original L-D-matching problem. Note that no cycle with size smaller than L is in the constructed graph. This property ensures that any matching on the graph is k-stable. ■

[6] The weight of one *edge* here refers to the sum of weights of the two directed arcs.

Algorithm 1: Cycle reduction algorithm

```
Input: The cycles in $L$-way exchange, $S_0$.
Output: A set of internally stable cycles $S$.
• $S \leftarrow S_0$
• While $\exists C$, such that $C$ is the longest cycle that is not
  internally stable among the cycles in $S$, do
    – Remove $C$ from $S$.
    – Let $T = (t_1, t_2, \ldots t_r)$ be a cycle satisfying
      $\forall i = 1, 2, \ldots, r$, $t_i \in C$, $t_i$ is allocated a better choice
      in $T$ than in $C$.
    – For each $t_i t_{i+1}$(subscripts mod. r) in $T$, $t_i t_{i+1}$ and
      part of $C$ forms a cycle $C_i$, add $C_i$ to $S$.
• Return $S$
```

L-k exchange in weighted graphs

Theorem 7 *For all $L > k \geq 3$, deciding whether social welfare can reach a constant c by L-k exchange in weighted graph is* NP-COMPLETE.

The proof can be adapted from the proof of Theorem 6, by assigning weights properly, which can be found in full version.

Theorem 8 *In weighted graphs, the worst case ratio of L-k exchange is $\frac{2}{n}$.*

Proof: **Upper bound.** First, the upper bound of the worst case ratio is $\frac{2}{n}$ as implied by the proof of Theorem 3.

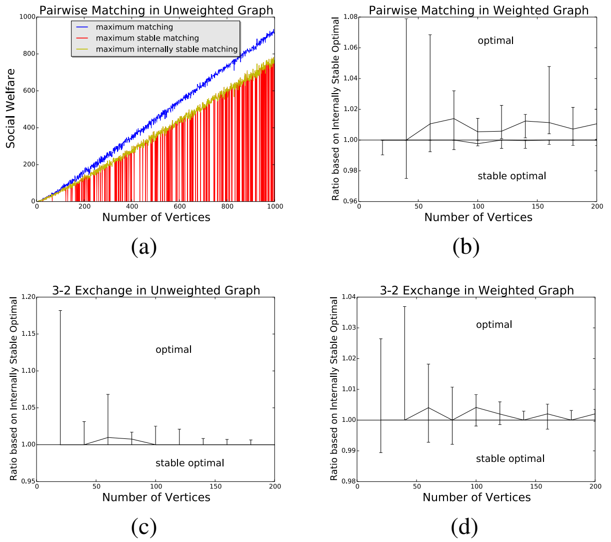
Tightness. We prove this by explicit constructing an internally stable matching via Algorithm 1.

The algorithm terminates because the C_i's generated from C have smaller lengths, so any removed cycle C won't appear again in S. This algorithm has two properties: (1) each arc in some element of S_0 will be in some element of S; (2) for any arc $v_i v_j$ in some element of S, if v_i points to v_k in some element of S_0, $w_{ij} \geq w_{ik}$.

Based on the properties above, we give a construction of internally stable matching that reaches ratio $\frac{n}{2}$. First, find $\arg_{v_x v_y} \max_{v_x v_y \in C \in S_0}(w_{xy} + w_{yz})$, z is the vertex y points to in S_0. Find $v_i v_j \in C^* \in S$, $w_{xy} + w_{yz}$ is more than $\frac{2}{n}$ of maximum L-way exchange. The total weight of C^* is not less than $w_{xy} + w_{yz}$. So, C^* is an internally stable matching with ratio at least $\frac{2}{n}$. ■

Experimental results

In this section, we evaluate the welfare loss caused by stability via experiments. All our experiments are conducted on the kidney exchange domain. We run experiments for all four cases considered in the paper. In each of these four cases, we compare the social welfare of the maximum unconstrained exchange and the maximum exchange under both notions of stability. Our data generator is carefully designed based on statistics of the US and China populations.(Tan, Zhou, and Tang 2006; Tu, Chen, and Wang 2005; Segev et al. 2005; Zhang 2004). We summarize our findings below. For the results on the China Data, see the full version.



(a) (b)

(c) (d)

Figure 4: U.S. data. In (b)(c)(d), the horizontal line denotes the welfare of internally stable optimal while the vertical lines above and below denote the range of the unconstrained optimal and stable optimal respectively. The curves connect the medians of unconstrained optimal.

- For pairwise matching in unweighted graphs, there is a clear gap between the unconstrained and internally stable optimal. Stable matching doesn't exist in many instances.
- For the other three cases, unconstrained optimal still beats internally stable optimal, which in turn beats stable optimal. However, the gaps are not significant.
- In all four cases, odd cycles return 0 for optimal stable matching. But such cases are unlikely to occur except for pairwise matching in unweighted graphs.
- We observe that the social welfare always grow linearly with the number of vertices. A regression analysis that returns the coefficients can be found in the full version.
- Internally stable exchange returns faster than the standard stability under our implementation because it is possible for CPLEX to cut more branches under internal stability.

Of independent interests, we include in the full version our ILP implementation for computing 3-2 exchanges.

Conclusion

In this paper, we study an alternative notion of stability called internal stability. We show desirable properties under this new notion: welfare loss and computation complexity. Our experiments show that by adding internally stability constraint, social welfare does not drop significantly under the setting of kidney exchange. All our findings suggest that internal stability is an efficient and robust trade-off point between exchange market design with and without stability constraints.

References

Abraham, D. J.; Blum, A.; and Sandholm, T. 2007. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM conference on Electronic commerce*, 295–304. ACM.

Anshelevich, E.; Chhabra, M.; Das, S.; and Gerrior, M. 2013. On the social welfare of mechanisms for repeated batch matching. In *AAAI*.

Anshelevich, E.; Das, S.; and Naamad, Y. 2013. Anarchy, stability, and utopia: creating better matchings. *Autonomous Agents and Multi-Agent Systems* 26(1):120–140.

Ashlagi, I., and Roth, A. 2011. Individual rationality and participation in large scale, multi-hospital kidney exchange. In *Proceedings of the 12th ACM conference on Electronic commerce*, 321–322. ACM.

Ashlagi, I.; Fischer, F. A.; Kash, I. A.; and Procaccia, A. D. 2010. Mix and match. In *ACM EC*, 305–314.

Awasthi, P., and Sandholm, T. 2009. Online stochastic optimization in the large: Application to kidney exchange. In *IJCAI*, volume 9, 405–411.

Banerjee, S.; Konishi, H.; and Sönmez, T. 2001. Core in a simple coalition formation game. *Social Choice and Welfare* 18(1):135–153.

Biró, P., and McDermid, E. 2010. Three-sided stable matchings with cyclic preferences. *Algorithmica* 58(1):5–18.

Bogomolnaia, A., and Jackson, M. O. 2002. The stability of hedonic coalition structures. *Games and Economic Behavior* 38(2):201–230.

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2012a. Dynamic matching via weighted myopia with application to kidney exchange. In *AAAI*.

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2012b. Optimizing kidney exchange with transplant chains: theory and reality. In *AAMAS*, 711–718.

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2013. Failure-aware kidney exchange. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, 323–340. ACM.

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2014. Price of fairness in kidney exchange. In *AAMAS, to appear*.

Dimitrov, D.; Borm, P.; Hendrickx, R.; and Sung, S. C. 2006. Simple priorities and core stability in hedonic games. *Social Choice and Welfare* 26(2):421–433.

Feder, T. 1992. A new fixed point approach for stable networks and stable marriages. *Journal of Computer and System Sciences* 45(2):233–284.

Gale, D., and Shapley, L. 1962. College admissions and the stability of marriage. *American Mathematical Monthly* 69(1):9–15.

Galil, Z. 1986. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)* 18(1):23–38.

Garey, M. R., and Johnson, D. S. 1979. Computers and intractability. a guide to the theory of np-completeness. a series of books in the mathematical sciences.

Huang, C.-C. 2010. Circular stable matching and 3-way kidney transplant. *Algorithmica* 58(1):137–150.

Irving, R. W. 1985. An efficient algorithm for the stable roommates problem. *Journal of Algorithms* 6(4):577–595.

Parkes, D., and Seuken, S. 2014. *Economics and Computation*. Cambridge University Press.

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

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

Roth, A. E. 2000. Game theory as a tool for market design. In *Game practice: Contributions from applied game theory*. Springer. 7–18.

Segev, D. L.; Gentry, S. E.; Warren, D. S.; Reeb, B.; and Montgomery, R. A. 2005. Kidney paired donation and optimizing the use of live donor organs. *JAMA: the journal of the American Medical Association* 293(15):1883–1890.

Tan, J.-m.; Zhou, Y.-c.; and Tang, X.-d. 2006. *Tissue Typing Technology and Clinical Application*. People's Medical Publishing House.

Tan, J. J. 1990. A maximum stable matching for the roommates problem. *BIT Numerical Mathematics* 30(4):631–640.

Tu, B.; Chen, J.-g.; and Wang, F. 2005. The distribution of abo bloodtype in kidney transplantation cases. *Journal of Chinese eugenics and heredity* 13(15):110–111.

Zhang, Z. 2004. Clinical study of renal issue typing. In *the collection of doctoral thesis*. Huazhong University of Science and Technology Press.