Service Exchange Problem

Julien Lesca[1], **Taiki Todo**[2]
[1] Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, 75016 Paris, France
[2] Department of Informatics, Kyushu University, Motooka 744, Fukuoka, Japan
julien.lesca@dauphine.fr, todo@inf.kyushu-u.ac.jp

Abstract

In this paper, we study the *service exchange problem* where each agent is willing to provide her service in order to receive in exchange the service of someone else. We assume that agent's preference depends both on the service that she receives and the person who receives her service. This framework is an extension of the housing market problem to preferences including a degree of externalities. We investigate the complexity of computing an individually rational and Pareto efficient allocation of services to agents for ordinal preferences, and the complexity of computing an allocation which maximizes either the utility sum or the utility of the least served agent for cardinal preferences.

1 Introduction

Finding a matching between indivisible items and agents has been widely studied in economics, and recently attracts considerable attention in AI/CS areas. When agents are initially equipped with some items, it is called as *exchange problem*. Housing market [Shapley and Scarf, 1974] is a special case where each agent is equipped with an item and has a linear priority over items. Various features of exchange problem have been studied, such as the strict core [Ma, 1994], externalities and indifferences [Mumcu and Saglam, 2007; Aziz and de Keijzer, 2012], and multiple endowments [Todo *et al.*, 2014; Sonoda *et al.*, 2014; Sikdar *et al.*, 2017].

In this paper we study an extension of housing market, called *service exchange problem*, where an agent's preference depends both on the item/service she receives and the person receiving her service. For example, consider a person who wants to study Dutch and can teach French. Assume she has two colleagues who can teach Dutch and want to study French. Whichever colleague she would choose as her Dutch teacher, she should also become her French teacher in return. Her preference then depends also on her student; she would prefer the one who is best for studying French, to minimize her cost. This model can reflect various practical situations, such as short-term house swap where customers prefer lending their houses to those they know well, and student-exchange between laboratories where a laboratory prefers to send its members to its privileged partners.

In our definition of the service exchange problem, the structure of externalities is very restricted; an agent is indifferent to exchanges not involving her. It is easy to see that the necessary condition by Sönmez (1999) for preference domains to guarantee the existence of truthful, Pareto efficient (PE), and individually rational (IR) matching rule does not hold for the service exchange problem. Therefore, our main focus in this paper is to find PE and IR matchings.

PE and IR matchings obviously exist by definition, not only in the service exchange problem but also in a broad class of exchange problems, and can be trivially found by enumerating and comparing all possible matchings. Therefore, a more precise description of our main focus is to check whether there is a computationally-efficient algorithm that finds a PE and IR matching. Many works have studied such algorithms [Fujita *et al.*, 2015; Aziz *et al.*, 2016; Gourvès *et al.*, 2017]. To the best of our knowledge, however, no work has focused on the service exchange problem.

Besides Pareto efficiency, there are various criteria for evaluating the matchings, especially when agents preferences are represented by cardinal utilities. One natural objective is to maximize the sum of agents' utilities, which reflects the perspective of utilitarianism. Another well-studied objective is to maximize the utility of the agent who has the minimum utility, which reflects the perspective of egalitarianism and is expected to result in a fair matching. Although both are very familiar in the literature of resource allocation [Chevaleyre *et al.*, 2006; Bouveret *et al.*, 2016], there has been no such discussion for the service exchange problem.

Our contribution is summarized as follows. For finding a PE and IR matching, we provide a poly-time algorithm for a special class of preferences, called *set-restricted*, and we show that the problem is NP-hard for general preferences. For Sum and Min objectives under cardinal utilities, we showed that maximizing either of them is NP-hard. Note that we do not address agents' incentive for manipulation, including manipulation complexity, i.e., a well-known alternative when truthfulness is not achievable [Bartholdi *et al.*, 1989; Teo *et al.*, 2001; Pini *et al.*, 2011; Fujita *et al.*, 2015]. Nevertheless, we believe that our contribution is critical; finding a desirable matching is likely to be hard even in the simple model where endowments are restricted to singletons and only a special type of externalities is allowed.

2 Preliminaries

Let $\mathcal{N} = \{1, \dots, n\}$ denote the set of agents, and let $\mathcal{S} = \{s_1, \dots, s_n\}$ denote the set of services, where s_i is the service offered by agent i. For any $N \subseteq \mathcal{N}$, let $\mathcal{S}_N$ denote the set of services belonging to the agents of N. A matching M is a bijection from $\mathcal{N}$ to $\mathcal{S}$, such that $M(i)$ is the service assigned to agent i in M. To simplify notations, we denote by $M(s_j)$ the agent receiving service s_j in M. Let $\mathcal{M}$ denote the set of possible matchings, and for any $N \subseteq \mathcal{N}$, let $\mathcal{M}_N$ denote the subset of $\mathcal{M}$ where each agent of N receives the service of an agent of N i.e., if $M \in \mathcal{M}_N$ then $\forall i \in N$, $M(s_i) \in N$. Note that $\mathcal{M}_N = \mathcal{M}_{\mathcal{N} \setminus N}$ holds for any $N \subseteq \mathcal{N}$.

The preference of an agent depends on both the service that she receives and the agent that receives her service. The ordinal preference of agent i is described through linear order $\succ_i$ on $\mathcal{N} \times \mathcal{S}$, where $(\ell, s_j) \succ_i (\ell', s_{j'})$ stands for agent i prefers receiving service s_j and serving agent ℓ rather than receiving service $s_{j'}$ and serving agent ℓ'. In the following, we call (i, s_i) the initial endowment of agent i. Preference $\succ_i$ is *set-restricted* (SR) if there exist sets $\mathcal{N}_i \subseteq \mathcal{N}$ and $\mathcal{S}_i \subseteq \mathcal{S}$ such that $(\ell, s_j) \succ_i (i, s_i)$ if and only if $\ell \in \mathcal{N}_i$ and $s_j \in \mathcal{S}_i$. In other words, $\mathcal{N}_i$ is the subset of agents that agent i accepts to serve, and $\mathcal{S}_i$ is the subset of services that agent i accepts to receive in exchange of her own service. Preference $\succ_i$ of agent i over matchings extends naturally: $M \succ_i M'$ if and only if $(M(s_i), M(i)) \succ_i (M'(s_i), M'(i))$. Notation $a \succsim_i b$ stands for $a \succ_i b$ or $a = b$.

In case of cardinal preferences, utility function $u_i : \mathcal{N} \times \mathcal{S} \to \mathbb{R}$ describes the preference of agent i as follows. For any pairs $(\ell, s_j), (\ell', s_{j'}) \in \mathcal{N} \times \mathcal{S}$, $(\ell, s_j) \succ_i (\ell', s_{j'})$ if and only if $u_i(\ell, s_j) > u_i(\ell', s_{j'})$. Utility function u_i is *additively separable* (AS) if exist functions $v_i : \mathcal{N} \to \mathbb{R}$ and $w_i : \mathcal{S} \to \mathbb{R}$ such that $u_i(j, s_\ell) = v_i(j) + w_i(s_\ell)$ holds for any $(j, s_\ell) \in \mathcal{N} \times \mathcal{S}$. We denote by $u(M) := (u_1(M), \dots, u_n(M))$ the vector of utility values achieved by matching M, where $u_i(M) := u_i(M(s_i), M(i))$.

For any coalition $N \subseteq \mathcal{N}$, matching M is Pareto dominated over N by matching M' if $M' \succsim_i M$ holds for any $i \in N$, and exists $j \in N$ such that $M' \succ_j M$. Matching M^* is *Pareto efficient* (PE) if no matching Pareto dominates it over $\mathcal{N}$. Matching M is *individually rational* (IR) if $(M(s_i), M(i)) \succsim_i (i, s_i)$ holds for any $i \in \mathcal{N}$. For cardinal preferences, social welfare function $f : \mathbb{R}^n \to \mathbb{R}$ can be used to compare vectors of utilities. Matching M^* is f-optimal if it maximizes $f(u(M))$. We consider in this paper two different social welfare functions: Sum (i.e., $f(u(M)) = \sum_{i \in \mathcal{N}} u_i(M)$) and Min (i.e., $f(u(M)) = \min_{i \in \mathcal{N}} u_i(M)$).

Example 1. *For $n = 3$, assume that ordinal preferences are $(3, s_3) \succ_1 (2, s_2) \succ_1 (2, s_3) \succ_1 (3, s_2), (1, s_3) \succ_2 (1, s_1)$ and $(1, s_1) \succ_3 (2, s_1)$, where only pairs strictly preferred to the initial endowment are represented. Note that preferences are SR with $\mathcal{N}_1 = \{2, 3\}, \mathcal{S}_1 = \{s_2, s_3\}, \mathcal{N}_2 = \{1\}, \mathcal{S}_2 = \{s_1, s_3\}, \mathcal{N}_3 = \{1, 2\}$ and $\mathcal{S}_3 = \{s_1\}$. Note also that no additively separable utility function can represent the preference of agent 1. Consider matching M_1, M_2, M_3 such that $M_1(1) = s_3$, $M_1(2) = s_2$, $M_1(3) = s_1$, $M_2(1) = s_2$, $M_2(2) = s_3$, $M_2(3) = s_1$ and M_3 is the initial endowment. All three are IR but only M_1 and M_2 are PE since M_3 is Pareto dominated over $\mathcal{N}$ by M_2.*

3 PE and IR for Ordinal Preferences

In this section, we study the complexity of computing a PE and IR matching for ordinal preferences. We start our analysis in Section 3.1 by assuming that all preferences are SR, and we provide a polynomial algorithm to compute a PE and IR matching. In section 3.2, we show that computing a PE and IR matching is in the general case NP-hard.

3.1 SR Preferences

We assume in this subsection that all preferences are SR and we describe an algorithm to compute a PE and IR matching. An *improving cycle* is a sequence of agents $\langle a_1, \dots, a_k \rangle$ such that $(a_{i-1}, s_{a_{i+1}}) \succ_{a_i} (a_i, s_{a_i})$ holds for any $i \in \{1, \dots, k\}$, where a_0 and a_{k+1} stand for a_k and a_1, respectively. If $N \subseteq \mathcal{N}$ is a superset of $\{a_1, \dots, a_k\}$ then $\langle a_1, \dots, a_k \rangle$ is an improving cycle restricted to N. Matching M *results* from improving cycle $\langle a_1, \dots, a_k \rangle$ if $M(a_i) = s_{a_{i+1}}$ holds for any $i \in \{1, \dots, k\}$, where s_{k+1} stands for s_1. Note that each agent visited by an augmenting cycle receives a strictly better outcome than her initial endowment in any matching resulting from this augmenting cycle.

We focus on *lexicographic improving cycles* which are improving cycles such that no other improving cycle provides a better outcome to the first agent in the sequence, and given that outcome for the first agent there is no other improving cycle providing a better outcome to the second agent, and so on. More formally, improving cycle $\langle a_1, \dots, a_k \rangle$ is a lexicographic improving cycle restricted to N if $\{a_1, \dots, a_k\} \subseteq N$ and there is no improving cycle $\langle a_1, a'_2, a'_3 \dots, a'_r \rangle$ restricted to N such that either $(a'_r, s_{a'_2}) \succ_{a_1} (a_k, s_{a_2})$, or $a'_r = a_k$ and there exists $\ell \in \{1, \dots, k\}$ such that $a'_i = a_i$ holds for any $i < \ell$, and $(a_{\ell-1}, s_{a'_{\ell+1}}) \succ_{a_\ell} (a_{\ell-1}, s_{a_{\ell+1}})$. Note that the definition relies on the starting point of the lexicographic comparison (denoted a_1 above). We assume in the following that the first agent in the sequence is this starting point. Note also that the lexicographic improving cycle starting from a given agent is unique since preferences are strict.

Algorithm 1 computes a PE and IR matching. It constructs step-by-step a matching by iteratively searching a lexicographic improving cycle starting from an arbitrary agent and by computing the matching resulting from this cycle.

Example 2. *We use the problem described in Example 1 to illustrate Algorithm 1. Assume that the agent picked during the first iteration is agent 1. There are three improving cycles visiting agent 1: $\langle 1, 3 \rangle, \langle 1, 2 \rangle$ and $\langle 1, 2, 3 \rangle$. $\langle 1, 3 \rangle$ is the lexicographic improving cycle when agent 1 is the starting point. Therefore, s_3 is assigned to agent 1 and s_1 is assigned to agent 3 in M^*. During the second iteration, only agent 2 remains in set A and no improving cycle exists. Therefore, agent 2 receives s_2 in M^* and the algorithm ends.*

Note that if agent 2 is picked during the first iteration of Algorithm 1, the improving cycles visiting agent 2 are $\langle 2, 3, 1 \rangle$ and $\langle 2, 1 \rangle$, and $\langle 2, 3, 1 \rangle$ is the lexicographic improving cycle. Therefore, Algorithm 1 returns $M^* = M_2$.

Its correctness is shown in the following proposition.

Algorithm 1

Require: Preference profile $(\succ_1, \ldots, \succ_n)$ of SR preferences
Ensure: A PE and IR matching

1: $A \leftarrow \mathcal{N}$.
2: $M^* \leftarrow \emptyset$.
3: **while** $A \neq \emptyset$ **do**
4:     Pick arbitrarily agent a_1 from A.
5:     Compute the lexicographic improving cycle $\langle a_1, \ldots, a_k \rangle$ restricted to A.
6:     **if** $k = 1$ **then** $\triangleright$ No improving cycle starts from a_1.
7:         $M^* \leftarrow M^* \cup \{(a_1, s_{a_1})\}$.
8:     **else**
9:         **for all** $j \in \{1, \ldots, k-1\}$ **do**
10:             $M^* \leftarrow M^* \cup \{(a_j, s_{a_{j+1}})\}$.
11:         $M^* \leftarrow M^* \cup \{(a_k, s_{a_1})\}$.
12:     $A \leftarrow A \setminus \{a_1, \ldots, a_k\}$.
13: **return** M^*.

Proposition 1. *Algorithm 1 computes a PE and IR matching.*

The proof relies on lemmas which appear in Appendix.

Proof. First of all, note that M^* is IR since either an agent receives her own service (line 7) or M^* results from an improving cycle visiting her (line 9-11). Let us show that M^* is not Pareto dominated by an IR matching of $\mathcal{M}$. With this in mind, we show by induction that M^* is not Pareto dominated over A_i by an IR matching of $\mathcal{M}_{A_i}$, where A_i represents set A at the beginning of iteration i of the **while** loop.

Base case: This case corresponds to the last iteration of the **while** loop. There are two possibilities: either $A_i = \{a_1\}$ or $A_i = \{a_1, \ldots, a_k\}$ with $k > 1$. In the first case, agent a_1 receives her own service in any matching of $\mathcal{M}_{A_i}$, and therefore none of them Pareto dominates M^* over A_i. In the latter case, $\langle a_1, \ldots, a_k \rangle$ is a lexicographic improving cycle restricted to A_i and M^* results from it. This implies by Lemma 1 that M^* is not Pareto dominated over A_i by an IR matching of $\mathcal{M}_{A_i}$.

Induction step: Assume that M^* is not Pareto dominated over A_{i+1} by an IR matching of $\mathcal{M}_{A_{i+1}}$, and let us show that M^* is not Pareto dominated over A_i by an IR matching of $\mathcal{M}_{A_i}$. Let N_1 denote the set of agents visited by the improving cycle computed in line 5. There are two possibilities: either $N_1 = \{a_1\}$ or $N_1 = \{a_1, \ldots, a_k\}$ with $k > 1$. In the first case, there is no improving cycle restricted to A_i visiting agent a_1. Therefore, M^* is not Pareto dominated over N_1 by an IR matching of $\mathcal{M}_{A_i}$ since a_1 must receive her own service in any IR matching of $\mathcal{M}_{A_i}$. In the latter case, $\langle a_1, \ldots, a_k \rangle$ is a lexicographic improving cycle restricted to A_i and M^* results from it. This implies by Lemma 1 that M^* is not Pareto dominated over N_1 by an IR matching of $\mathcal{M}_{A_i}$. In both cases, let N_2 denote $A_i \setminus N_1$. Note that $N_2 = A_{i+1}$ (line 12), and (N_1, N_2) forms a partition of A_i. We know that M^* is not Pareto dominated over N_2 by an IR matching of $\mathcal{M}_{N_2}$ (by induction hypothesis), and M^* is not Pareto dominated over N_1 by an IR matching of $\mathcal{M}_{A_i}$. This implies by Lemma 2 that M is not Pareto dominated over A_i by an IR matching of $\mathcal{M}_{A_i}$, which concludes the proof by induction.

For $i = 1$, this implies that M^* is not Pareto dominated over A_1 by an IR matching of $\mathcal{M}_{A_1}$. Therefore, M^* is not Pareto dominated by any IR matching since $A_1 = \mathcal{N}$ and $\mathcal{M}_{\mathcal{N}} = \mathcal{M}$. To conclude, let us show by contradiction that M^* is PE. Assume that M' is a matching Pareto dominating M^* over $\mathcal{N}$. This implies that M' is IR since M^* is IR and $M' \succsim_i M^*$ for any $i \in N$. This leads to a contradiction with M^* is not Pareto dominated by any IR matching. □

Algorithm 2

Require: Preference profile $(\succ_1, \ldots, \succ_n)$ of SR preferences, subset of agents $A \subseteq \mathcal{N}$, agent $a_1 \in A$
Ensure: A lexicographic improving cycle restricted to A

$N \leftarrow A \setminus \{a_1\}$
$P_{a_1} \leftarrow Pile(N, \succ_{a_1})$.
$(a_e, s_{a_2}) \leftarrow pop(P_{a_1})$.
while $noPath(a_2, a_e, G_N)$ and $P_{a_1} \neq \emptyset$ **do**
    $(a_e, s_{a_2}) \leftarrow pop(P_{a_1})$.
if $noPath(a_2, a_e, G_N)$ **then**
    return $\langle a_1 \rangle$ $\triangleright$ No improving cycle starts from a_1.
$k \leftarrow 2$.
while $a_k \neq a_e$ **do**
    $N \leftarrow N \setminus \{a_k\}$
    $P_{a_k} \leftarrow Pile(N, \succ_{a_k}, a_{k-1})$.
    $s_{a_{k+1}} \leftarrow pop(P_{a_k})$.
    while $noPath(a_{k+1}, a_e, G_N)$ **do**
        $s_{a_{k+1}} \leftarrow pop(P_{a_k})$.
    $k \leftarrow k + 1$.
return $\langle a_1, \ldots, a_k \rangle$.

It remains to show that Algorithm 1 can run in polynomial time. The size of A decreases during each **while** loop (line 12). Therefore, the number of iterations of the **while** loop is bounded by n. Hence, Algorithm 1 is polynomial if a lexicographic improving cycle restricted to A can be computed in polynomial time. Algorithm 2 performs this task by constructing, agent by agent, the lexicographic improving cycle. In the first **while** loop, it computes the most preferred pair of agent/service for agent a_1 which leads to an improving cycle. To do so, it uses two data structures: pile P_{a_1} and graph G_N.

After its construction by function $Pile$, P_{a_1} contains all pairs (i, s_j) of $N \times \mathcal{S}_N$ such that $(i, s_j) \succ_{a_1} (a_1, s_{a_1})$ and $a_1 \in \mathcal{S}_i$, ordered from the most preferred to the least preferred according to $\succ_{a_1}$. Function *pop* returns the first pair of P_{a_1} and removes it from P_{a_1}. Oriented graph G_N is used to check the existence of an improving cycle starting from a given agent. Its set of vertices is N and its set of edges is $\{(i, j) \in N^2 : s_j \in \mathcal{S}_i, i \in \mathcal{N}_j\}$. Note that for any edge (i, j), s_j can be assigned to agent i without violating IR. Therefore, if (a_e, s_{a_2}) belongs to P_{a_1} and exists path $a_2 \rightarrow \ldots \rightarrow a_e$ in G_N then $\langle a_1, a_2, \ldots, a_e \rangle$ is an improving cycle restricted to $N \cup \{a_1\}$. Function $noPath(a, b, G_N)$ returns true if no path from a to b exists in G_N.

At the end of the first **while** loop, either no improving cycle is founded and $\langle a_1 \rangle$ is returned, or (a_e, s_{a_2}) is the most preferred pair for agent a_1 in any improving cycle restricted to A. In the latter case, Algorithm 2 completes the sequence by computing, agent by agent, the most preferred remaining service for the last agent in the sequence when she serves her predecessor in the sequence. Note that N is updated after each step to reflect the set of remaining services. After its construction by function $Pile$, P_{a_k} contains all services s_i of $\mathcal{S}_{a_k} \cap N$ such that $a_k \in \mathcal{N}_i$, ordered form the most preferred to the least preferred for agent a_k when she serves agent a_{k-1}.

Example 3. *We use the problem described in Example 1 to illustrate Algorithm 2. Assume that $A = \{1, 2, 3\}$ and $a_1 = 2$. N is set to $\{1, 3\}$, pile P_{a_1} contains both $(1, s_3)$ and $(1, s_1)$, and $G_{\{1,3\}}$ is described in Figure 1. The first pair considered in P_{a_1} is $(1, s_3)$, and a_e and a_2 are set to 1 and 3, respectively. There is a path from 3 to 1 in $G_{\{1,3\}}$, and therefore the algorithm does not enter in the first **while** loop. Since $a_2 \neq a_e$, the algorithm enter in the second **while** loop. N is set to $\{1\}$, pile P_{a_2} only contains service s_1, and $G_{\{1\}}$ is a single vertex. Then, a_3 is set to 1. There is obviously a path from 1 to 1 and therefore the algorithm does not enter in the inner **while** loop. The second **while** loop ends since $a_3 = a_e$, and lexicographic improving cycle $\langle 2, 3, 1\rangle$ is returned.*

Algorithm 2 runs in polynomial time since the sizes of piles P_{a_1} and P_{a_k} are bounded above by n^2 and the size of N decreases after each iteration of the second **while** loop. The following proposition shows the correctness of Algorithm 2.

Proposition 2. *Algorithm 2 computes a lexicographic improving cycle starting from a_1 and restricted to A.*

Proof. Algorithm 2 either returns *(i)* $\langle a_1\rangle$ or *(ii)* $\langle a_1, \dots, a_k\rangle$. In case *(i)*, assume by contradiction that exists improving cycle $\langle a_1, a'_2, \dots, a'_r\rangle$ restricted to A. Since preferences are SR, $a_1 \in \mathcal{S}_{a'_r}$, $(a'_r, s_{a'_2}) \succ_{a_1} (a_1, s_{a_1})$, $s_{a'_i} \in \mathcal{S}_{a'_{i-1}}$ and $a'_{i-1} \in \mathcal{N}_{a'_i}$ hold for any $i \in \{3, \dots, r\}$. Hence, $a'_2 \to \dots \to a'_r$ is a path in G_N. Therefore, $(a'_2, s_{a'_r})$ is contained in P_{a_1} and $noPath(a'_2, a'_r, G_N) = false$, leading to a contradiction with $noPath(a_2, a_e, G_N) = true$ at the end of the first **while** loop (line 6 in Algorithm 2).

In case *(ii)*, the first **while** loop stops with $noPath(a_2, a_e, G_N) = false$. This implies that there exists a path $a_2 \to a'_3 \to \dots \to a_e$ in G_N. Furthermore, we know by definition of P_{a_1} that $(a_e, s_{a_2}) \succ_{a_1} (a_1, s_{a_1})$ and $a_1 \in \mathcal{S}_{a_e}$. Therefore, there exists at least one improving cycle $\langle a_1, a_2, a'_3, \dots, a_e\rangle$. Note that $(a_1, s_{a'_3}) \succ_{a_2} (a_2, s_{a_2})$ and $a_2 \in \mathcal{N}_{a'_3}$ hold. Hence, during the first iteration of the second **while** loop, the inner **while** loop (lines 13-14) stops with a_3 such that $(a_1, s_{a_3}) \succ_{a_2} (a_2, s_{a_2})$. By applying the same reasoning over the sequence, one can check that for any $i \in \{3, \dots, k\}$, $(a_{i-1}, s_{a_{i+1}}) \succ_{a_i} (a_i, s_{a_i})$ holds. Therefore, $\langle a_1, \dots, a_k\rangle$ is an improving cycle.

Assume by contradiction that exists improving cycle $\langle a_1, a'_2, a'_3 \dots, , a'_r\rangle$ restricted to A such that either $(a'_r, s_{a'_2}) \succ_{a_1} (a_k, s_{a_2})$, or $a'_r = a_k$ and there exists $\ell \in \{1, \dots, k\}$ such that $a'_i = a_i$ holds for any $i < \ell$, and $(a_{\ell-1}, s_{a'_{\ell+1}}) \succ_{a_\ell} (a_{\ell-1}, s_{a_{\ell+1}})$. In the former case, pair $(a'_r, s_{a'_2})$ belongs to P_{a_1} and is tested before (a_k, s_{a_2}) in the first **while** loop. Therefore, $noPath(a'_2, a'_r, G_N)$ should be true, a contradiction with $\langle a_1, a'_2, a'_3 \dots, a'_r\rangle$ is an improving cycle. In the latter case, path $a'_{\ell+1} \to \dots \to a'_r$ exists in G_N for $N = A \setminus \{a_1, \dots, a_\ell\}$ since $\langle a_1, \dots, a_\ell, a'_{\ell+1} \dots, a'_r\rangle$ is an improving cycle. Furthermore, we know that $a'_r = a_k = a_e$. On the other hand, $a'_{\ell+1}$ should be tested before $a_{\ell+1}$ in the inner **while** loop and $noPath(a'_{\ell+1}, a_e, G_N)$ necessarily returns true, leading to a contradiction. □

Note that the correctness of Algorithm 1 does not rely on the assumption of SR preferences and still holds in the general case. However, the search for a lexicographic improving cycle is NP-hard in the general case whereas it can be computed in polynomial time with the assumption of SR preferences by Algorithm 2.

3.2 General Preferences

Algorithm 1 applies to general preferences whereas Algorithm 2 is restricted to SR preferences. Is it possible to extends Algorithm 2 to general preferences? The following proposition provides a negative answer to this question.

Proposition 3. *Computing a PE and IR matching is NP-hard.*

Proof. We present a reduction from the NP-complete problem (3,B2)-SAT [Berman and Karpinski, 2003]. An instance of (3,B2)-SAT contains a set of variables $\mathcal{X} = \{x_1, \dots, x_s\}$ and a set of clauses $\mathcal{C} = \{c_1, \dots, c_t\}$. Each clause has three literals, and each variable appears in exactly 4 clauses of $\mathcal{C}$, twice as a positive literal and twice as a negative literal. The problem is to decide whether there exists a truth assignment of $\mathcal{X}$ such that all clauses in $\mathcal{C}$ are true.

The reduction is as follows. For each variable x_i, we introduce 4 literal-agents X_i^1, X_i^2, $\bar{X}_i^1$ and $\bar{X}_i^2$. Agents X_i^1 and X_i^2 represent occurrences of x_i as positive literals, and agents $\bar{X}_i^1$ and $\bar{X}_i^2$ represent occurrences of x_i as negative literals. Furthermore, for each clause c_i, we introduce clause-agent C_i, and an additional dummy clause-agent C_0. For notational convenience, we denote by L_i^j the literal-agent corresponding to the literal appearing at position j in c_i, and we denote by $\ell(i, j)$ and $\bar{\ell}(i, j)$ the indices of the clauses where appear occurrence j of literals x_i and $\neg x_i$, respectively. Furthermore, for any subsets $A \subseteq \mathcal{N}$ and $B \subseteq \mathcal{S}$, we denote by $[A \times B]$ an arbitrary order over pairs in $\{(a, b) : a \in A, b \in B\}$. Finally, for any $1 < i \leq s$, any $1 \leq j < t$ and any $1 \leq k < t$, preferences are as follows. Note that only preferences over the pairs preferred to the initial endowment are provided.

$$
\begin{aligned}
X_1^1 &: (C_{\ell(1,1)-1}, s_{C_{\ell(1,1)}}) \succ (C_t, s_{X_1^2})\\
\bar{X}_1^1 &: (C_{\bar{\ell}(1,1)-1}, s_{C_{\bar{\ell}(1,1)}}) \succ (C_t, s_{\bar{X}_1^2})\\
X_i^1 &: (C_{\ell(i,1)-1}, s_{C_{\ell(i,1)}}) \succ \left[\{X_{i-1}^2, \bar{X}_{i-1}^2\} \times \{s_{X_i^2}\}\right]\\
\bar{X}_i^1 &: (C_{\bar{\ell}(i,1)-1}, s_{C_{\bar{\ell}(i,1)}}) \succ \left[\{X_{i-1}^2, \bar{X}_{i-1}^2\} \times \{s_{\bar{X}_i^2}\}\right]\\
X_j^2 &: (C_{\ell(j,2)-1}, s_{C_{\ell(j,2)}}) \succ \left[\{X_j^1\} \times \{s_{X_{j+1}^1}, s_{\bar{X}_{j+1}^1}\}\right]\\
\bar{X}_j^2 &: (C_{\bar{\ell}(j,2)-1}, s_{C_{\bar{\ell}(j,2)}}) \succ \left[\{\bar{X}_j^1\} \times \{s_{X_{j+1}^1}, s_{\bar{X}_{j+1}^1}\}\right]\\
X_t^2 &: (C_{\ell(t,2)-1}, s_{C_{\ell(t,2)}}) \succ (X_t^1, s_{C_0})\\
\bar{X}_t^2 &: (C_{\bar{\ell}(t,2)-1}, s_{C_{\bar{\ell}(t,2)}}) \succ (\bar{X}_t^1, s_{C_0})\\
C_0 &: \left[\{X_t^2, \bar{X}_t^2\} \times \{s_{L_1^1}, s_{L_1^2}, s_{L_1^3}\}\right]\\
C_k &: \left[\{L_k^1, L_k^2, L_k^3\} \times \{s_{L_{k+1}^1}, s_{L_{k+1}^2}, s_{L_{k+1}^3}\}\right]\\
C_t &: \left[\{L_t^1, L_t^2, L_t^3\} \times \{s_{X_1^1}, s_{\bar{X}_1^1}\}\right]
\end{aligned}
$$

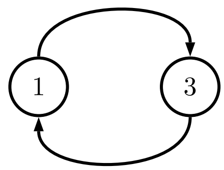


Figure 1: Graph $G_{\{1,3\}}$

Informally, preferences are designed such that there exists an improving cycle iff $\mathcal{C}$ is satisfiable. Any improving cycle can be divided into two parts. The first part is a path alternating between clause-agents and literal-agents, starting from C_0 and ending at C_t. In this path, the literal-agent between clause-agents C_k and C_{k+1} is a literal of c_k. The second part of the cycle is a path which visits for each variable x_i, either both X_i^1 and X_i^2 or both $\bar{X}_i^1$ and $\bar{X}_i^2$. The second part of the cycle insure that no two literal-agents visited in the first part of the cycle correspond to a literal and its negation.

We show that there exists a matching Pareto dominating the initial endowment over $\mathcal{N}$ iff the instance of $\mathcal{C}$ is satisfiable. Assume that there exists truth assignment φ of $\mathcal{X}$ such that each clause in $\mathcal{C}$ is true. We construct from φ a matching which Pareto dominates over $\mathcal{N}$ the initial endowment. Let $L_i^{j_i}$ be a literal-agent corresponding to a literal of c_i which is true according to φ. The services of $L_i^{j_i}$ and C_i are assigned to agents C_{i-1} and $L_i^{j_i}$, respectively. For each variable x_i, if x_i is set to true in φ then assign the service of $\bar{X}_i^2$ to agent $\bar{X}_i^1$, and depending on the truth value of x_{i-1}, assign the service of $\bar{X}_i^1$ either to agent C_t if $i = 1$, either to agent X_{i-1}^2 if $x_{i-1} = false$, or to agent $\bar{X}_{i-1}^2$ if $x_{i-1} = true$. If x_i is set to false in φ then assign the service of X_i^2 to agents X_i^1, and depending on the truth value of x_{i-1}, assign the service of X_i^1 either to agent C_t if $i = 1$, either to agent X_{i-1}^2 if $x_{i-1} = false$, or to agent $\bar{X}_{i-1}^2$ if $x_{i-1} = true$. The other agents are assigned to their own services. It is easy to check that this assignment is IR and Pareto dominates the initial endowment over $\mathcal{N}$.

The proof of the other implication relies on the structure of the preferences described above. The formal proof of this implication is omitted due to space limitation. □

4 Sum and Min for Cardinal Preferences

In this section, we study the complexity of computing Sum-optimal and Min-optimal matchings for cardinal preferences.

4.1 Sum for General and AS Preferences

We first show that the computation of a Sum-optimal matching is hard to solve for general cardinal preferences.

Corollary 1. *Computing Sum-optimal matching is NP-hard.*

Proof. The reduction is essentially the same than the one provided in the proof of Proposition 3. We only need to give the utility values defining the preferences of the different agents. For each agent in this reduction, assign utility $8t$ to her initial endowment, and utility $8t+1$ for any pair which is preferred to her initial endowment. Other pairs have utility value 0. This utility values enforce any matching which maximizes the sum to be IR. Indeed, the utility sum of any matching which provides utility 0 to at least one agent is bounded above by $4t(8t+1) = 32t^2 + 4t$, whereas the utility sum for the initial endowment is $(4t+1)(8t) = 32t^2 + 8t$. Therefore, checking the existence of a matching which Pareto dominates the initial endowment over $\mathcal{N}$ amounts to decide if there exists a matching of utility sum strictly greater than $32t^2 + 4t$. Note that the utility values implies indifferences since two pairs can have the same utility value. However, it is possible to slightly perturb the utility values by adding very small values in order to remove indifferences. □

The remaining of the subsection is devoted to the description of a polynomial algorithm to compute a Sum-optimal matching for AS preferences. This algorithm relies on the well known assignment problem which a restriction of our setting where utility functions only depend on the service assigned to an agent. More formally, the utility function $u'_i : \mathcal{N} \times \mathcal{S} \to \mathbb{R}$ of agent i is such that $u'_i(j, s_\ell) = w'_i(s_\ell)$ holds for any $j, \ell \in \mathcal{N}$. The assignment problem aims to find a Sum-optimal matching for utility functions u'. It is solvable in polynomial time by the Hungarian method [Kuhn, 1955].

We reduce the computation of a Sum-optimal matching, in an instance of the service exchange problem with cardinal AS preferences u, to an assignment problem with utility function u'. The set of agents and the set of services remain the same. Note that utility function u'_i of agent i can be described through w'_i. For any agents $i, \ell \in \mathcal{N}$, we define the value of w'_i over s_ℓ as $w'_i(s_\ell) = w_i(s_\ell) + v_\ell(i)$. To conclude, we show that the utility sum of matching M is the same for utility functions u and $u' = (u'_1, \ldots, u'_n)$. By definition of u', we have $\sum_{i\in\mathcal{N}} u'_i(M) = \sum_{i\in\mathcal{N}}[w_i(M(i)) + v_{S(i)}(i)] = \sum_{i\in\mathcal{N}}[w_i(M(i)) + v_i(M(s_i))] = \sum_{i\in\mathcal{N}} u_i(M)$, where $S(i)$ denotes the owner of $M(i)$. Therefore, the optimal matchings for this two problems are the same.

Note that this approach can be applied to the search of a Sum-optimal and IR matching for SR and AS preferences. Indeed, it is possible to consider in an assignment problem that some services are unacceptable to an agent and they cannot be matched. By requiring in the assignment problem that the set of acceptable services for agent i is $\{s_j : i \in \mathcal{N}_j \wedge s_j \in \mathcal{S}_i\}$, we insure that any complete matching is IR. Therefore, it is possible to compute a Sum-optimal matching among the set of IR matchings. Then, by comparing its value with the value of the Sum-optimal matching among the full set of matchings, one can conclude on the existence of an IR and Sum-optimal matching: these values coincide if and only if a Sum-optimal and IR matching exists.

4.2 Min for AS Preferences

The following proposition shows that even for restricted preferences, computing a Min-optimal matching is hard.

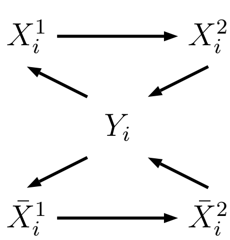
Proposition 4. *Computing a Min-optimal matching is NP-hard even for AS utility functions.*

Proof. We present a reduction from (3,B2)-SAT. This problem is formally described in the proof of Proposition 3.

The reduction is as follows. For each variable x_i, we introduce a gadget containing 5 agents X_i^1, X_i^2, Y_i, $\bar{X}_i^1$ and $\bar{X}_i^2$. Agents X_i^1 and X_i^2 represent occurrences of x_i as positive literals, and agents $\bar{X}_i^1$ and $\bar{X}_i^2$ represent occurrences of x_i as negative literals. Agents X_i^1, X_i^2, $\bar{X}_i^1$ and $\bar{X}_i^2$ are called literal-agents. Non zero utility values inside the gadget are depicted in Figure 2. Intuitively speaking, this gadget forces agent Y_i to exchange her service according to one of the two improving cycles appearing in this figure. Hence, either Y_i exchanges her service with literal-agents X_i^1 and X_i^2, or she exchanges her service with literal-agents $\bar{X}_i^1$ and $\bar{X}_i^2$. For each clause c_i we introduce agent C_i. Agent C_i has utility 1 for receiving a service from a literal-agent which corresponds to a literal of c_i, and she has utility 2 for serving this literal-agent. Symmetrically, each literal-agent corresponding to a literal of c_i has utility 1 for receiving the service of C_i, and utility 2 for serving C_i. Intuitively speaking, agent C_i will perform a one-to-one exchange with a literal-agent corresponding to one of its literals in order to achieve utility 3. Finally, for each $i \in \{1, \ldots, 2s - t\}$, we introduce agent Z_i. These agents play the role of garbage collectors for unassigned literal-agents. Agent Z_i has utility 1 for receiving a service from a literal-agent, and utility 2 for serving a literal-agent. Symmetrically, each literal agent has utility 1 for receiving a service from Z_i, and utility 2 for serving Z_i.

Let us show that $\mathcal{C}$ is satisfiable if and only if there exists a matching providing an utility of at least 3 to each agent. Assume that there exists a truth assignment φ of $\mathcal{X}$ that satisfies $\mathcal{C}$. For each variable x_i, if x_i is true in φ then assign services according to improving cycle $\langle Y_i, \bar{X}_i^1, \bar{X}_i^2\rangle$, and otherwise assign services according to improving cycle $\langle Y_i, X_i^1, X_i^2\rangle$. For each clause c_i, choose one literal which is true according to φ and perform a one-to-one exchange of services between the corresponding literal-agent and C_i. Finally, each literal-agent who did not receive a service so far performs a one-to-one exchange with some agent Z_i. It is easy to check that each agent achieves an utility of 3 in this matching.

Assume that there exists a matching M which provides an utility of at least 3 to each agent. We construct from M a truth assignment φ of the variables in $\mathcal{X}$. Let us first focus on agent Y_i. Her preference are such that she receives the service of either X_i^1 or $\bar{X}_i^1$ to achieve an utility of 3. Assume that Y_i receives the service of X_i^1 and let us show that M should result from improving cycle $\langle Y_i, X_i^1, X_i^2\rangle$. If X_i^1 serves Y_i (with utility 1) then she should receive $s_{X_i^2}$ because it is the only service providing her utility 2. Then, since X_i^2 serves X_i^1 (with utility 1), X_i^2 should receive s_{Y_i} because it is the only services providing her utility 2. Following the same line of reasoning, we can conclude that if agent Y_i receives $s_{\bar{X}_i^1}$ then agent $\bar{X}_i^1$ should receive $s_{\bar{X}_i^2}$ and agent $\bar{X}_i^2$ should receive s_{Y_i}. Hence, for each variable x_i, M results from either $\langle Y_i, X_i^1, X_i^2\rangle$ or $\langle Y_i, \bar{X}_i^1, \bar{X}_i^2\rangle$. In the former case, set variable x_i to false in φ and in the latter case, set variable x_i to true in φ. Furthermore, for achieving an utility of 3 in M, agent C_i must obtain the service of an agent corresponding to a literal of c_i. Since this literal-agent was not involve in one of the improving cycles describer above, the truth value of its corresponding variable satisfies c_i. Therefore, φ satisfies all clauses of $\mathcal{C}$. □



$$
\begin{aligned}
&v_{X_i^1}(Y_i) = v_{\bar{X}_i^1}(Y_i) = 1\\
&w_{X_i^1}(s_{X_i^2}) = w_{\bar{X}_i^1}(s_{\bar{X}_i^2}) = 2\\
&v_{X_i^2}(X_i^1) = v_{\bar{X}_i^2}(\bar{X}_i^1) = 1\\
&w_{X_i^2}(s_{Y_i}) = w_{\bar{X}_i^2}(s_{Y_i}) = 2\\
&v_{Y_i}(X_i^2) = v_{Y_i}(\bar{X}_i^2) = 1\\
&w_{Y_i}(s_{X_i^1}) = w_{Y_i}(s_{\bar{X}_i^1}) = 2
\end{aligned}
$$

Figure 2: Gadget for variable x_i

5 Conclusion and Future Works

We introduced the service exchange problem where each agent can provide and receive a single service, and where preferences are defined over pairs costumer/service. We studied the computational complexity of finding a PE and IR, Sum-optimal and Min-optimal matchings under different preference restrictions. Our main achievement is a polynomial algorithm to compute a PE and IR for SR preferences.

An interesting line of research would be to extend the polynomial algorithm for computing a PE and IR matching to a broader class of preferences. Another direction is the computation of the core. It would be also interesting to study the complexity of computing a Sum-optimal and IR matching for AS cardinal preferences without SR preferences assumption. Finally, even if most of the problems studied in this paper are computationally hard, it would be worthy to consider MIP formulations of these problems able to solve instances of realistic size (e.g., [Lesca and Perny, 2010; Lesca *et al.*, 2013]).

Acknowledgments

Taiki Todo thanks JSPS for support under JSPS KAKENHI Grant Numbers JP17H04695 and JP17H00761. Julien Lesca acknowledges support from the ANR project CoCoRICo-CoDec, as well as the JCJC project MAPPOLEON.

A Appendix

The following lemma shows that a matching resulting from a lexicographic improving cycle possesses some guarantees of performance regarding Pareto dominance.

Lemma 1. *If matching M results from lexicographic improving cycle $\langle a_1, \ldots, a_k\rangle$ restricted to N then M is not Pareto dominated over $\{a_1, \ldots, a_k\}$ by an IR matching of $\mathcal{M}_N$.*

Proof. By contradiction, assume that M' is an IR matching of $\mathcal{M}_N$ which Pareto dominates M over $\{a_1, \ldots, a_k\}$. Let ℓ be the smallest integer such that $(M'(s_{a_\ell}), M'(a_\ell)) \succ_{a_\ell} (M(s_{a_\ell}), M(a_\ell))$ holds. Since M' belongs to $\mathcal{M}_N$ and is IR, M' results from an improving cycle $\langle a_1, \ldots, a_\ell, a'_{\ell+1}, \ldots, a'_r\rangle$ restricted to N. The existence of this improving cycle is a contradiction with $\langle a_1, \ldots, a_k\rangle$ is a lexicographic improving cycle restricted to N. □

The following lemma shows that Pareto dominance over two disjoint sets can imply Pareto dominance over its union.

Lemma 2. *For any $N \subseteq \mathcal{N}$, any partition (N_1, N_2) of N, and any matching M of $\mathcal{M}_{N_1}$, if M is both not Pareto dominated over N_1 by an IR matching of $\mathcal{M}_N$, and not Pareto dominated over N_2 by an IR matching of $\mathcal{M}_{N_2}$ then M is not Pareto dominated over N by an IR matching of $\mathcal{M}_N$.*

Proof. Assume by contradiction that M is both not Pareto dominated over N_1 by an IR matching of $\mathcal{M}_N$, and not Pareto dominated over N_2 by an IR matching of $\mathcal{M}_{N_2}$, and M' is an IR matching of $\mathcal{M}_N$ which Pareto dominates M over N. This means that $M' \succsim_i M$ holds for any $i \in N$, and exists $j \in N$ such that $M' \succ_j M$ holds. If exists $j \in N_1$ such that $M' \succ_j M$ then M' Pareto dominates M over N_1, leading to a contradiction since M' is an IR matching of $\mathcal{M}_N$. Therefore, $M \succsim_i M'$ holds for any $i \in N_1$. This implies that $M(i) = M'(i)$ holds for any $i \in N_1$ since both $M \succsim_i M'$ and $M' \succsim_i M$ hold and there is no indifference. Therefore, M' belongs to $\mathcal{M}_{N_2}$ since both $M \in \mathcal{M}_{N_1}$ and $M' \in \mathcal{M}_N$ hold. M' cannot Pareto dominate M over N_2 since M' is an IR matching of $\mathcal{M}_{N_2}$. But in that case, M' neither Pareto dominates M over N_1 nor N_2, leading to a contradiction with M' Pareto dominate M over N. □

References

[Aziz and de Keijzer, 2012] Haris Aziz and Bart de Keijzer. Housing markets with indifferences: A tale of two mechanisms. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.

[Aziz *et al.*, 2016] Haris Aziz, Péter Biró, Jérôme Lang, Julien Lesca, and Jérôme Monnot. Optimal reallocation under additive and ordinal preferences. In *Proceedings of the 15th International Conference on Autonomous Agents & Multiagent Systems*, pages 402–410, 2016.

[Bartholdi *et al.*, 1989] III Bartholdi, J.J., C.A. Tovey, and M.A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[Berman and Karpinski, 2003] Piotr Berman and Marek Karpinski. Improved approximation lower bounds on small occurrence optimization. *Electronic Colloquium on Computational Complexity (ECCC)*, 10(008), 2003.

[Bouveret *et al.*, 2016] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. Fair allocation of indivisible goods. In *Handbook of Computational Social Choice*, pages 284–310. 2016.

[Chevaleyre *et al.*, 2006] Yann Chevaleyre, Paul E Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaitre, Nicolas Maudet, Julian Padget, Steve Phelps, Juan A Rodriguez-Aguilar, and Paulo Sousa. Issues in multiagent resource allocation. *Informatica*, 30(1), 2006.

[Fujita *et al.*, 2015] Etsushi Fujita, Julien Lesca, Akihisa Sonoda, Taiki Todo, and Makoto Yokoo. A complexity approach for core-selecting exchange with multiple indivisible goods under lexicographic preferences. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 907–913, 2015.

[Gourvès *et al.*, 2017] Laurent Gourvès, Julien Lesca, and Anaëlle Wilczynski. Object allocation via swaps along a social network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 213–219, 2017.

[Kuhn, 1955] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[Lesca and Perny, 2010] Julien Lesca and Patrice Perny. LP solvable models for multiagent fair allocation problems. In *Proceedings of the 19th European Conference on Artificial Intelligence*, pages 387–392, 2010.

[Lesca *et al.*, 2013] Julien Lesca, Michel Minoux, and Patrice Perny. Compact versus noncompact LP formulations for minimizing convex Choquet integrals. *Discrete Applied Mathematics*, 161(1-2):184–199, 2013.

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

[Mumcu and Saglam, 2007] Ayse Mumcu and Ismail Saglam. The core of a housing market with externalities. *Economics Bulletin*, 3(57):1–5, 2007.

[Pini *et al.*, 2011] Maria Silvia Pini, Francesca Rossi, K. Brent Venable, and Toby Walsh. Manipulation complexity and gender neutrality in stable marriage procedures. *Autonomous Agents and Multi-Agent Systems*, 22(1):183–199, 2011.

[Shapley and Scarf, 1974] Lloyd Shapley and Herbert Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37, 1974.

[Sikdar *et al.*, 2017] Sujoy Sikdar, Sibel Adali, and Lirong Xia. Mechanism design for multi-type housing markets. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 684–690, 2017.

[Sönmez, 1999] Tayfun Sönmez. Strategy-proofness and essentially single-valued cores. *Econometrica*, 67(3):677–689, 1999.

[Sonoda *et al.*, 2014] Akihisa Sonoda, Etsushi Fujita, Taiki Todo, and Makoto Yokoo. Two case studies for trading multiple indivisible goods with indifferences. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 791–797, 2014.

[Teo *et al.*, 2001] Chung-Piaw Teo, Jay Sethuraman, and Wee-Peng Tan. Gale-Shapley stable marriage problem revisited: Strategic issues and applications. *Management Science*, 47(9):1252–1267, September 2001.

[Todo *et al.*, 2014] Taiki Todo, Haixin Sun, and Makoto Yokoo. Strategyproof exchange with multiple private endowments. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 805–811, 2014.