Randomized Assignments for Barter Exchanges: Fairness vs Efficiency*

Wenyi Fang[1], Aris Filos-Ratsikas[2], Søren Kristoffer Stiil Frederiksen[2], Pingzhong Tang[1], and Song Zuo[1]

[1] Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
{fwy13@mails, zuos13@mails}@mails.tsinghua.edu.cn,
kenshin@tsinghua.edu.cn

[2] Department of Computer Science, Aarhus University, Aarhus, Denmark
{filosra,ssf}@cs.au.dk

Abstract. We study fairness and efficiency properties of randomized algorithms for barter exchanges with direct applications to kidney exchange problems. It is well documented that randomization can serve as a tool to ensure fairness among participants. However, in many applications, practical constraints often restrict the maximum allowed cycle-length of the exchange and for randomized algorithms, this imposes constraints of the cycle-length of every realized exchange in their decomposition. We prove that standard fairness properties such as envy-freeness or symmetry are incompatible with even the weakest notion of economic efficiency in this setting. On the plus side, we adapt some well-known matching mechanisms to incorporate the restricted cycle constraint and evaluate their performance experimentally on instances of the kidney exchange problem, showing tradeoffs between fairness and efficiency.

1 Introduction

Over the past years, barter exchanges, with kidney exchange as a representative example, have become a topic of intensive research at the intersection of AI and economics. In a barter exchange, participants enter the system with some endowment and then exchange their endowments in order to obtain better allocations. Such exchanges are very popular in settings such as exchanges of used

* 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, a Tsinghua Initiative Scientific Research Grant and a China Youth 1000-talent program. Aris Filos-Ratsikas and Søren Kristoffer Stiil Frederiksen acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which this work was performed. The authors also acknowledge support from the Center for Research in Foundations of Electronic Markets (CFEM), supported by the Danish Strategic Research Council.

books or DVDs where agents do not have very high values for their endowments and would rather trade them with others.

The current literature on barter exchanges presents two major challenges, *fairness* and *implementability* [13,2,3,7,8,9,6]. Most would agree that a fair procedure should guarantee at least properties like *symmetry* and *envy-freeness*. In practical applications, barter exchanges tend to be carried out *deterministically*; on the other hand as argued in a number of papers [5,14,11], central economic notions of fairness such as the ones stated above, require randomization.

Implementability has to do with whether the designed exchanges can be carried out in practice. The simplest exchanges are *pairwise* exchanges, involving only two agents [14,12]. Exchanges can also be more complicated, involving multiple participants, exchanging endowments in a cycle. A vital constraint of most such exchange systems (kidney exchanges, room exchanges[3]) is that the number of agents involved in a cycle must be bounded [14,2]. Such constraints may be imposed for a number of reasons; a real-life motivating example comes from perhaps the most widespread applications of barter exchanges, the *kidney exchange problem*.

In a kidney exchange market, pairs consisting of incompatible donors and patients enter the market, in search for other pairs to exchange kidneys with. In case of inter-pair compatibility, i.e. when the donor of the first pair is compatible with the patient of the second pair and vice-versa, an exchange is carried out. In many countries, such exchange systems have been in effect for several years.

There are several constraints on the length of such exchanges however, imposed for both practical and ethical reasons. First of all, participants involved in an exchange cycle must conduct surgery simultaneously at the same hospital[4], making it logistically infeasible for any hospital to host a large cycle. Secondly, donors can not be contractually obligated to donate their kidneys, since it is illegal in most countries. If an "offline" exchange were to take place, there is no guarantee that donors would not opt out after their counterparts receive their transplants. [8].

For this reason, the length of the exchange cycles is constrained to be a small number (three in most cases). This however, makes the problem much more challenging. It is known that, under such constraints on the cycle-length, the problem of finding an efficient exchange is NP-hard [2]. Despite the theoretical hardness results, Abraham et al. [2] designed an algorithm that through several optimization techniques produces an optimal exchange on typical instances of the problem in reasonable running time. The algorithm, while efficient, is deterministic and not tailored to incorporate fairness criteria. On the other hand, fairness is a key property here; between patients with similar compatibility characteristics and similar needs, no deterministic choice can be justified, especially if it results in loss of human life. In fact, Dickerson et. al. [8] observe that, among

[3] `http://reslife.umd.edu/housing/reassignments/roomexchange/`

[4] In this paper, we do not consider the use of altruistic chains, which may circumvent this requirement.

the exchanges in the major organ exchange system in the United States, UNOS,[5] only 7 percent of them finally make it to surgery. The lack of fairness guarantees might be a possible explanation for this phenomenon.

As mentioned earlier, randomization can be used as a way of achieving fairness and there are many candidate mechanisms in matching and exchange literature to choose from. Perhaps the two best-studied are the Probabilistic Serial mechanism [5] and Random Serial Dictatorship [1]. Both achieve fairness in the sense of symmetry but the former is also envy-free. However, the implementability constraint introduces complications to the use of randomized mechanisms as well. Given the natural interpretation of a randomized exchange as a probability mixture over deterministic exchanges, the constraint requires that *every* such exchange does not contain long cycles. If we restrict the possible outcomes to those assignments only, it is unclear whether the mechanisms maintain any of their fairness properties. This is one of the questions that we address in this paper.

We investigate the problem of designing barter exchanges that meet the two desiderata above. In particular, we explore the use of randomized mechanisms for achieving tradeoffs between efficiency and fairness, under the added constraint of restricted cycle-length in their decomposition. We make the following two-fold contribution.

- First, we consider the tradeoffs between economic efficiency and fairness from a theoretical point of view and prove that even the weakest form of economic efficiency (a relaxation of ex-post Pareto efficiency that is suitable for the problem) is incompatible with both *envy-freeness* and *symmetry*. On the other hand, we show that it is possible to satisfy each property independently, together with the restricted cycle-length.
- Next, we adapt two well-known mechanisms, *Random Serial Dictatorship* and *Probabilistic Serial* to incorporate the cycle-length constraint and evaluate their performance on instances of the kidney exchange problem. We show tradeoffs between the efficiency (in the sense of social welfare) and quantified envy-freeness of the exchange for those adaptations of the mechanisms and compare them to the mechanism that produces an optimal assignment.

Most relevant to the current paper is the work by Balbuzanov [4], where the author considers deterministic and randomized mechanisms for barter exchanges, under the restriction of the cycle length in the components of the decomposition, very similarly to what we do here. Interestingly, he presents an adaptation of the Probabilistic Serial mechanism (named the "2-cycle Probabilistic Serial") that always produces components with cycles of length two and satisfies two desired properties: *ordinal efficiency* (a stronger notion of efficiency than the one we consider here) and anonymity, i.e. a guarantee that the outcome is imprevious to renaming agent/item pairs. It is well-known that ordinal efficiency implies ex-post Pareto efficiency; in particular this is also true for the relaxed versions of efficiency that we consider in this paper. Furthermore, anonymity (together with

[5] `www.unos.org`.

neutrality) is known to imply symmetry. The existence of 2-cycle Probabilistic Serial however does not contradict our negative result on compatibility between ex-post efficiency and symmetry; crucially, the anonymity notion used in [4] does not imply symmetry in our setting.

2 Background

Let $N = \{1, \ldots, n\}$ be a set of agents and let $M = \{1, \ldots, n\}$ be a set of items. We assume that each agent is associated with exactly one item and without loss of generality, let agent i be associated with item i. Let item i be the *endowment* of agent i. Each agent has *valuations* over the items., i.e. numerical values that denote her levels of satisfaction. Let $\mathbf{v_i} = (v_{i1}, \ldots, v_{in})$ be the *valuation vector* of agent i and let $\mathbf{V} = (\mathbf{v_1} \ldots \mathbf{v_n})$ be a *valuation matrix*.

An *assignment* D is a matching of agents to items, such that each agent receives exactly one item. This is precisely a permutation matrix, where entry $d_{ij} = 1$ if agent i receives item j and 0 otherwise. Alternatively, one can view D as a directed graph $D = (N, E)$, where a vertex v_i corresponds to both agent and item i and an edge (i, j) means that agent i is matched with item j in D. Given this interpretation, an assignment is a set of disjoint cycles, where agents exchange endowments along a cycle. If the maximum length of any cycle in such an assignment is k, we will say that the assignment is *k-restricted*.[6]

Since each agent receives exactly one item in expectation, a *probabilistic* or *randomized* assignment is a bistochastic matrix P where entry p_{ij} denotes the probability that agent i is matched with item j. We will call $p_i = (p_{i1}, \ldots, p_{in})$ an *assignment vector*. A *mechanism* is a function that on input a valuation matrix V outputs an assignment P.

A probabilistic assignment P can be viewed as a probability mixture over deterministic assignments. This is due to the *Birkhoff-von Neumann* theorem that states that each bistochastic matrix of size n can be written as a convex combination of at most n^2 permutation matrices. Since it is particularly relevant to the design of our mechanisms, we will describe the decomposition process in more detail in Section 4. We will say that a randomized assignment P is k-restricted if it can be written as a probability mixture of k-restricted deterministic assignments.

The two standard notions of fairness that we consider in this paper are *envy-freeness* and *symmetry*. An assignment is (ex-ante) envy-free if no agent would prefer to swap assignment vectors with any other agent. An assignment is *symmetric* if all agents that have identical valuation vectors receive identical assignment vectors. Note that envy-freeness does not imply symmetry; two agents with identical valuations could be equally satisfied with an assignment without having the same probabilities of receiving each item[7].

[6] Balbuzanov [4] uses the term *k-constrained* to describe such assignments.

[7] This is true in particular because we consider all mechanisms, including *cardinal* mechanisms, i.e. mechanisms that can use the numerical values when outputting assignments.

A deterministic assignment D is *Pareto efficient* if there is no other assignment D' that is not less preferable for any agent and strictly more preferable for at least one agent. If such an assignment D' exists, we will say that D' *Pareto dominates* D.

It is not hard to see that standard efficiency is not compatible with k-restricted assignments. To see this, let $n > k$ and let V be such that for $i = 1, \ldots, n-1$, $\max_{j \in M} v_{ij} = i + 1$ and $\max_{j \in M} v_{nj} = 1$. Any ex-post Pareto efficient assignment must consist single permutation matrix D consisting of only one cycle of length n, which is of course not a k-restricted assignment. For that reason, it makes sense to only define Pareto efficiency in terms of k-restricted assignments.

Definition 1. *A deterministic assignment D is Pareto efficient if there does not exist any k-restricted assignment that Pareto dominates it.*

Remark 1. Note that this is the same definition of *k-constrained Pareto efficiency* used in [4]. For simplicity and since the only notion of efficiency in this paper is with respect to k-restricted assignments, we simply use the term Pareto-efficiency.

For randomized mechanisms, an assignment is *ex-ante* Pareto efficient, if there is no other assignment that satisfies the condition above in expectation. An assignment is *ex-post* Pareto efficient if for every realization of randomness, there is no other assignment that Pareto dominates it. In other words, a randomized assignment is ex-post Pareto efficient if it can be written as a probability mixture of Pareto efficient deterministic assignments. Note that ex-post Pareto efficiency is the weakest efficiency notion for randomized mechanisms in literature.

3 Fairness and economic efficiency

In this section, we explore the compatibility and incompatibility between fairness and efficiency properties given the constraint of small cycles. We show that even the weakest form of efficiency is incompatible with two well-known fairness criteria, even in the case when $k = 3$. Note that $k = 3$ is the common choice for the maximum allowed cycle length in the most important applications of the problem, that of kidney exchange that we discuss in Section 5.

Theorem 1. *There is no mechanism that always outputs an assignment which is envy-free, ex-post Pareto efficient and 3-restricted.*

Proof. First, we prove the theorem for $n = 4$. First we construct the valuation profile $\mathbf{v}$ with $\mathbf{v_1} = (22, 27, 81, 79)$, $\mathbf{v_2} = (14, 67, 36, 16)$, $\mathbf{v_3} = (48, 6, 33, 88)$ and $\mathbf{v_4} = (36, 87, 91, 90)$. Next, we generate all possible permutation matrices with four elements and eliminate those that contain cycles of length more than 3. Then also remove those that are not Pareto efficient according to the preference orderings profile induced by $\mathbf{v}$. Let $D = \{D_1, D_2, \ldots, D_{|D|}\}$ be the resulting set of permutation matrices. Then, we need to solve the following constraint satisfaction problem:

Find $P, \alpha_1, \ldots, \alpha_{|D|}$ such that

$$\text{(i)} \ \sum_j p_{kj} v_{kj} \geq \sum_j p_{lj} v_{kj} \ \forall k, l, \qquad \textit{(envy-freeness)}$$

$$\text{(ii)} \ P = \sum_{i=1}^{|D|} \alpha_i D_i, \qquad \textit{(decomposition)}$$

$$\text{(iii)} \ \sum_{i=1}^{n} p_{ij} = \sum_{j=1}^{n} p_{ij} = 1, \qquad \textit{(valid assignment)}$$

$$\text{(iv)} \ \alpha_i \geq 0 \ \forall i, \qquad \textit{(valid coefficients)}$$

$$\text{(v)} \ 0 \leq p_{ij} \leq 1 \ \forall i, j. \qquad \textit{(valid probabilities)}$$

For the valuation profile that we created, the constraint satisfaction problem is infeasible. This proves the theorem for $n = 4$. We will use that as the base case to prove that the theorem is true for any n using induction on the size of the valuation matrix.

For size $k-1$, there is no envy-free, ex-post Pareto efficient mechanism that produces a 3-restricted assignment by the induction hypothesis. Let V_{k-1} be the valuation matrix for size $k-1$ and let U_k be the matrix obtained by V_{k-1} by adding agent k and item k such that for all agents $i \neq k$, $v_{ij} > v_{ik}$ for all items $j \neq k$ and $v_{kk} > v_{jk}$ for all items $j \neq k$. Then, any ex-post Pareto efficient mechanism on input V_k must allocate item k to agent k with probability 1, otherwise there would be some Pareto-dominated permutation matrix in the decomposition. If such a mechanism existed, the assignment of the remaining $k-1$ items to the remaining $k-1$ agents would imply the existence of an envy free, ex-post Pareto efficient mechanism for input V_{k-1}, contradicting the induction hypothesis. □

Next, we prove a similar impossibility theorem for the case when the notions of fairness is symmetry.

Theorem 2. *There is no mechanism that always outputs an assignment which is symmetric, ex-post Pareto efficient and* 3*-restricted.*

Proof. The proof idea is similar to that of the proof of Theorem 1, the main difference being that in the constraint satisfaction problem, the constraint for envy-freeness is replaced by:

$$P_{ik} = P_{jk}, \forall k \in M, \forall i, j \in N \text{ such that } v_{il} = v_{jl} \ \ \forall l \in M,$$

which is the constraint for symmetry. For the new constraint satisfaction problem, for $n = 5$, if we let $\mathbf{v_1} = \mathbf{v_2} = (4, 3, 5, 1, 2)$, $\mathbf{v_3} = \mathbf{v_4} = \mathbf{v_5} = (2, 5, 1, 4, 3)$, then the problem is infeasible. To extend the theorem for any n, we can use exactly the same inductive argument we use in the proof of Theorem 1. □

Remark 2. As we mention in the introduction, Balbuzanov proposes the *2-cycle Probabilistic Serial* mechanism, which is ex-ante Pareto efficient and anonymous. This does not contradict Theorem 2 because his definition of anonymity is with respect to pairs of agents and endowed items and does not imply symmetry. [8]

Next, we prove that both ex-post Pareto efficiency and envy-freeness or symmetry are needed for the impossibilities. If we remove ex-post efficiency, the simple mechanism that allocates all items uniformly at random, which is trivially both envy-free and symmetric, also produces k-restricted assignments.

Theorem 3. *The mechanism* $\mathrm{U_n}$ *that always outputs a uniform random assignment always produces a* k*-restricted assignment for* $k \geq 2$*. Furthermore, if* n *is odd, then the decomposition of the assignment consists of* n *permutation matrices, each one of which contains a self-loop and* $\frac{n-1}{2}$ *pairs. If* n *is even, then the decomposition consists of* n *permutation matrices,* $n-1$ *of which contain* $\frac{n}{2}$ *pairs and one which contains* n *self-loops.*

Proof. We will consider the cases when n is odd and n is even separately. Since all entries p_{ij} of the assignment matrix are $1/n$, the coefficients of the decomposition will be $1/n$ and for any i and j, d_{ij} will be 1 in exactly one component D and 0 in all others. Assume first that n is odd. Recall the graph interpretation of D and observe that D is a regular $n-1$-sided polygon. For vertex i, let e_i be the opposite side of G to vertex i and let D_i be the permutation matrix consisting of the self-loop (i) and pairs (kl) where k and l are adjacent to e_i or adjacent to a diagonal parallel to e_i. To get the decomposition, we iterate over all $i = 1, \dots, n$ and obtain the permutation matrices D_i. Note that each permutation matrix consists of one self-loop and $\frac{n-1}{2}$ pairs. Next, assume that n is even, which means that $n-1$ is odd. Let $\mathrm{U_{n-1}}$ be the assignment matrix of size $n-1$ and $\mathrm{U_n}$ be the assignment matrix after we add agent n and item n. Since $n-1$ is odd, $\mathrm{U_{n-1}}$ can be decomposed into permutation matrices that contain one self-loop and $\frac{n-2}{2}$ pairs. The decomposition of $\mathrm{U_n}$ will be exactly the same, except that for each self-loop of each permutation matrix, we create a pair with item n, and we add an additional permutation matrix consisting only of self-loops. Again, it is not hard to see that the decomposition consists of n components, $n-1$ of which contain $\frac{n}{2}$ pairs and one that contains n self-loops. □

Finally, if we only require Pareto efficiency without any regard to fairness, it is trivial to obtain a deterministic Pareto efficient mechanism. The mechanism is the following simple one. Given an input valuation matrix V, generate all possible permutation matrices and find a feasible one that is Pareto efficient (with respect to the set of k-restricted components).

[8] A simple example with two agents 1, 2 that have the same preference over items 1, 2 is sufficient to see this.

4 Randomized Mechanisms

In this section, we design mechanisms that output k-restricted assignments. Recall that a randomized mechanism inputs a valuation profile (or a preference profile) for n agents and outputs a bistochastic assignment matrix P. The assignment P can then be decomposed into at most n^2 permutation matrices using the Birkhoff-von Neumann decomposition.

The Birkhoff-von Neumann decomposition

The decomposition works as follows. First, from P, construct a binary matrix P^{bin} by setting $p_{ij}^{\mathrm{bin}} = 1$ if $p_{ij} > 0$ and 0 otherwise. From P^{bin}, construct a bipartite graph G with vertices corresponding to the rows and the columns of P^{bin} and with edges corresponding to the non-zero entries of P^{bin}. In other words, edge (i,j) exists in G if and only if $p_{ij}^{\mathrm{bin}} = 1$. Using Hall's theorem, one can easily prove that G has a perfect matching. Note that this matching corresponds to some permutation matrix Π. Find such a D in G and find the smallest entry (i,j) in P such that $\Pi_{ij} = 1$ and let a be the value of that entry. For every entry (i,j) in P such that $\Pi_{ij} = 1$, subtract a from (i,j) to obtain a substochastic matrix P'. Then apply the same procedure again on P'. Note that a will be the coefficient of the first component of the decomposition and D will be that component. Also note that since P' has at least one more zero entry than P, the procedure will terminate in at most n^2 steps.

In our case, we are interested only in k-restricted components of the decomposition. One way to handle components with longer cycles is to remove them from the decomposition and redistribute their probabilities (given by their coefficients) to k-restricted components. It is conceivable that some of the properties of the assignment that are satisfied *in expectation* might be lost during the process; on the other hand, properties satisfied ex-post are preserved. To evaluate the ex-ante properties of the new assignment, we can re-construct the bistochastic matrix based on the components that survived the previous step. We will call this process the *recomposition* of the assignment matrix.

The process described above can be used to transform any mechanism to one that produces k-restricted assignments, assuming that the original decomposition had at least one k-restricted component. In general, this is not always the case however; it could be that some other decomposition (with possibly more than n^2 components) is needed in order to find such a component. Even worse, it could be the case that such a decomposition does not exist. We observe that in general, it is hard to decide whether this is the case or not.

Theorem 4. *Let P be an assignment. Deciding whether any decomposition of P has a 3-restricted component is NP-hard.*

Proof. Abraham et al. [2] proved that finding a cycle-cover consisting of cycles of length at most 3 is NP-hard. In their reduction, they use the gadget shown in Figure 4, also known as a *clamp*. They construct a graph where clamps only intersect with other clamps on vertices, x_a, y_b and z_c. To get some intuition about the construction, one can think as x_a, y_b and z_c as elements in sets X, Y and Z respectively. Let $T \subseteq X \cup Y \cup Z$ be a set of triples. Two clamps intersect at a vertex x_a if x_a is part of two different triples (x_a, y_b, z_c) and (x_a, y'_b, z'_c) [9]. We will refer to the subgraph consisting of vertices x_a, $1, \ldots, L-1$ (on the left in Figure 4), x_a^i and the edges indicident to them as "the x part of the clamp".

Recall the definition of graph D corresponding to the binary matrix P^{bin} at the Birkhoff-von Neumann decomposition described earlier; the graph has edges (i, j) only between vertices satisfying $P_{ij}^{\text{bin}} = 1$. We claim that there exists a decomposition of P with at least one 3-restricted component if and only if graph D has a cycle cover consisting of cycles of length at most 3. It is not hard to see that there exists a decomposition of P with at least one 3-restricted component if and only if Graph D has a cycle cover consisting of cycles of length at most 3. It then suffices to prove that the graph used in [2] corresponds to some binary matrix P^{bin} associated with a bistochastic matrix P.

To find such a matrix, it is enough to find an assignment of weights $p_1, \ldots, p_{|E|}$ to the edges of the graph, such that for every vertex, the total weight of incoming edges and the total weight of outgoing edges is 1; such a weight assignment corresponds directly to a bistochastic matrix. We will only specify the weights for edges in the x part of the clamp; the rest are defined symmetrically. Let s be the in-degree of x_a.

First, for edges $e = (1, 2), \ldots, (L-2, L-1)$, let $p_e = 1$.

Then let:

$$
\begin{aligned}
p_{(L-1,x_a)} &= \frac{1}{s}, & p_{(L-1,x_a^i)} &= \frac{s-1}{s}, \\
p_{(x_a,1)} &= \frac{1}{s}, & p_{(z_c^i,x_a^i)} &= \frac{1}{s}, \\
p_{(x_a^i,1)} &= \frac{s-1}{s}, & p_{(x_a^i,y_b^i)} &= \frac{s-1}{s}.
\end{aligned}
$$

It is not hard to see that the assigned weights satisfy the constraint above and hence correspond to some bistochastic matrix P. □

In the following, we describe a general method to generate k-restricted assignments based on some original assignment P. We will call this method *small-cycle projection.* Note that the decomposition-recomposition procedure that we described earlier can be viewed as such a projection.

Small-cycle Projection : Given a bistochastic matrix P and a "distance" measure d, the small-cycle projection P^* of P with respect to d is the solution of the following program,

$$
\begin{aligned}
&\text{minimize} \quad d(P^*, P) \\
&\text{subject to } P^* \in \text{Conv}(\mathcal{D}_k)
\end{aligned}
\tag{1}
$$

[9] This interpretation is very natural given that the proof in [2] uses a reduction from 3D-Matching.

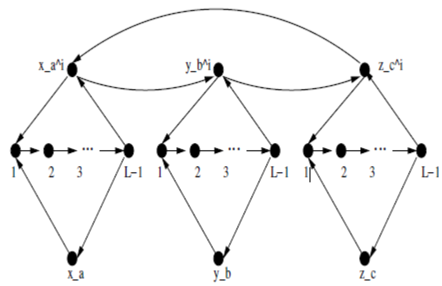

Fig. 1. The gadget used in in the proof of Theorem 4 as it appears in Abraham et al. [2].

where $\mathcal{D}_k$ is the set of all k-restricted deterministic assignments and $\mathrm{Conv}(\mathcal{D}_k)$ is the convex hull of $\mathcal{D}_k$.

A key observation here is that if d is a linear function, Program (1) is a linear program. Unfortunately, even if k is chosen to be 3, the set $\mathcal{D}_k$ is exponentially large in n. For this reason, we present two ways to approximate $\mathrm{Conv}(\mathcal{D}_k)$ via a small subset of $\mathcal{D}_k$.

Small-Cycle Projection via Randomized Birkhoff-von Neumann Decomposition

Recall that the Birkhoff-von Neumann decomposition operates by finding a perfect matching on a bipartite graph in each step. Using the decomposition-recomposition procedure, we can approximate P by the matrix composed of the k-restricted components of the decomposition. It is conceivable however that different decompositions yield different k-restricted components and the choice of decomposition plays a central role to the quality of the approximation. For this reason, we need to have some freedom to choose between decompositions. On the other hand, iterating over all decompositions is computationally intractable. To balance the need for flexibility and the computational burdens, we employ the algorithm proposed by Goel, Kapralov, and Khanna [10]. Their algorithm computes random perfect matchings which can then be used to obtain random decompositions. The mechanism is then simply:

(i) decompose P as $\sum_i \lambda_i D_i$;
(ii) recompose $P^* = \sum_{D_i \in \mathcal{D}_k} \lambda_i^* D_i$.

The redistribution of probabilities can be done in various ways; the simplest being equally among k-restricted components.

Small Cycle Projection via Sequential Randomized Small-cycle Cover

The second approach generates a set of random k-restricted permutations to approximate $\mathcal{D}_k$ in program (1). Particularly, these k-restricted permutations are generated by an algorithm that finds a maximum weight cycle cover consisting of cycles of length at most k, on Graph G corresponding to assignment P. For example, when $k = 3$, the algorithm by [2] can be used, where the weights are randomly assigned to the edges induced by the fractional allocation P. Formally, we describe our generating method in Algorithm 1.

Algorithm 1: Generating Small-cycle Permutations.

```
input  : P
output: D̂_k
D̂_k ← ∅;
while |D̂_k| < α(n) do
    for i, j in [n] do
        if P_ij = 0 then
            G_ij ← 0;
        else
            G_ij ← rand();
    D̂_k ← D̂_k ∪ {MaxWeightSmallCycleCover(G, k)};
return Ŝ_k
```

In the algorithm, $\alpha(n)$ is the desired size of set $\hat{\mathcal{D}}_k$, $\mathsf{rand}()$ generates a random number in $[1, 1+\epsilon]$, and finally MaxWeightSmallCycleCover(G, k) returns the maximum weighted cycle cover of length at most k in graph G.

Unlike the randomized Birkhoff-von Neumann decomposition, the k-restricted permutations generated here are not a decomposition of the input P. Hence we need to solve Program (1) with some properly chosen "distance measure" d to approximate the assignment P. For linear distance measures, the program can be easily solved.

In the remainder of the section, we adapt two well-known mechanisms, *Probabilistic Serial* and *Random Serial Dictatorship* to make them compatible with 3-restricted allocations, in order to use them in our experiments in Section 5. As mentioned earlier, the reason for the choice of $k = 3$ is because this is the standard maximum allowed cycle length in kidney exchange operations. For Probabilistic Serial, we apply the small-cycle projection method; for Random Serial Dictatorship, we apply a different construction that always admits a decomposition with small cycles.

4.1 Probabilistic Serial with restricted cycles

The Probabilistic Serial mechanism works as follows. Each item is interpreted as an infinitely divisible good that the agents consume over the unit interval $[0, 1]$ at the same fixed speed. Each agent starts consuming her favorite item until the item is entirely consumed. Then, she moves to the next item on her preference list that has not been entirely consumed and starts consuming it. The procedure terminates when all items are entirely consumed. The fraction p_{ij} of item j consumed by agent i is then interpreted as the probability of assigning item j to agent i.

Using the small-cycle projection methods described above, we can construct two variants of the mechanism. For the variants generated using the randomized small-cycle cover method, we choose two appropriate distance metrics as follows to generate the approximate assignment P^*.

- *Social welfare distance*: $d_{\text{welfare}}(P, P^*) = \langle \mathbf{V}, P - P^* \rangle$ measuring the difference of social welfares induced by P and P^*, where $\langle \cdot, \cdot \rangle$ is the pointwise product of matrices.
- l_∞ *distance*: $d_{\text{norm}}(P, P^*) = \max_{i,j} |P_{ij} - P^*_{ij}|$ measuing the l_∞ error of P^* approaching P.

4.2 Random Serial Cycle

Random Serial Cycle, or *RSC* for short is a straightforward adaptation of Random Serial Dictatorship to incorporate the short cycle constraint. Specifically, the mechanism first uniformly at random fixes an ordering of agents and then matches them serially with their favorite items from the set of available items, just like Random Serial Dictatorship does. The difference is that whenever the length of an exchange is $k-1$ and the exchange is not a cycle, the next agent to be picked is matched with the item that "closes" the cycle, regardless of her preferences. For example, for $k = 3$, if some agent i is matched with item j and agent j is matched with item l and agent l is next to pick an item, she will be matched with item i.

By construction, any run of the mechanism outputs a k-restricted assignment. To evaluate its properties however, we need to compose the assignment matrix from the probability mixture of the outcomes, which if done naively would require us to generate all possible $n!$ orderings of agents. In fact, it has been shown [15] that it is $\#P$-hard to compute the assignment matrix of *RSD* given an input valuation matrix.

To sidestep this complication, we modify the mechanism to instead generate n^2 orderings at random using a Monte-Carlo process. The details follow. First we explain how to generate the orderings that we use and then we describe the mechanism.

Orderings generation

- Fix a permutation π of $\{1, \dots, n\}$ uniformly at random from the set of all permutations with n elements. Initialize position i of order π to be 1.
- For each position i, swap the agent in position i, denoted by π_i, with randomly chosen from π_i to π_n (could be agent π_i as well). Increase i by 1 at each iteration.
- Repeat until $i = n$.

Random Serial Cycle mechanism

- Generate n^2 orders uniformly at random by Monte-Carlo process. Initialize the assignment matrix to be n-by-n zero matrix.
- For each order π, Initialize position i of order π to be 1. Agent π_i tries to be matched with the most preferred available item, denoted by $\mu(\pi_i)$, until reaches one of the following conditions:
  - Agent $\mu(\pi_i)$ ranks item π_i on top of her preference list and $\pi_i, \mu(\pi_i)$ are matched together.
  - Agent $\mu(\pi_i)$ finds an item j such that agent j has positive value for π_i after searching for the most preferred available item. Then the exchange cycle is completed.

  Remove these matched pairs from the ordering. If agent π_i does not find a qualified item $\mu(\pi_i)$, then she is matched to item π_i. Increase i by 1 at each iteration.
- When position i is bigger than n, the k-restricted permutation matrix for ordering π can be produced. Multiply it by $1/n^2$ and add it to allocation matrix.
- After processing all the orderings, the assignment matrix P is produced.

5 Experimental results

In this section, we design experiments to evaluate the performance of our mechanisms in terms of efficiency and fairness. Our experiments are conducted on the kidney exchange domain with input from realistic data. Our data generator is carefully designed based on statistics of the population in the United States. The generator incorporates patients and donors' physiological characteristics such as age, gender, blood type, HLA antigen, panel reactive antibodies (PRA) and number of waiting years prior to transplantation. It then produces a matching score for each patient-donor pair to quantify the transplantation quality. The input is then a weighted, directed, bipartite graph where the weight of an edge (i, j) is the utility (transplantation quality) of the patient of a donor-patient pair i when being matched with the donor of pair j. The transplantation quality can be interpreted as the probability that an exchange between the two participating pairs will be successful. For all of the experiments, k will be equal to 3.

Our measure of efficiency will be the *social welfare*, i.e. the sum of the transplantation qualities of the expected matching over all participating pairs. For fairness, we will try to minimize the fraction of envious agents, where the envy is calculated *in expectation*, i.e. an agent is envious if she would prefer another agents expected assignment to hers. We run experiments for different input sizes, ranging from a few agents to a hundred agents. Real-life input sizes can be larger but exchanges involving no more than a hundred agents are often carried out as well in practice. The bottleneck for the running time is the decomposition; as the number of agents grows larger, the harder it gets to achieve a decomposition with 3-restricted components.

We compare three mechanisms in terms of their social welfare and fraction of envy. The first one is the modification of Probabilistic Serial (PS) that we obtain by applying the randomized small-cycle cover method for small-cycle projection. We implement two variants of modified Probabilistic Serial, namely *PS-welfare* and *PS-norm*, based on the *social welfare distance* and the l_∞ *distance* respectively. The second mechanism that we use is Random Serial Cycle (RSC) with Monte-Carlo random generation of n^2 orderings of agents. Finally, we consider the optimal mechanism (denoted OPT in Figures 2 and 3) that computes the optimal social welfare of the exchange, by Abraham et al. [2]. The variant of Probabilistic Serial that we obtain from applying the randomized Birkhoff-von Neumann small-cycle projection turned out to be ineffective for our goals, the problem being that k-restricted components were not present in most decompositions, but it is conceivable that it could be effective using some other mechanism as the "basis" of the decomposition.

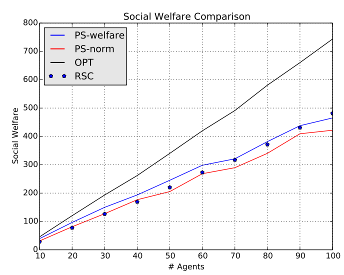

Fig. 2.

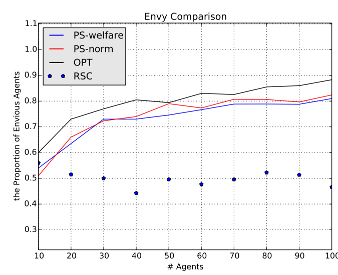

Fig. 3.

5.1 Fairness/Efficiency tradeoffs

In Figures 2 and 3, we show the results of the experiments for data from the U.S. population. As expected, the optimal mechanism performs better in terms of social welfare. The performance of our mechanisms (the two variants of PS and RSC) is very similar and not too far from the performance of the optimal mechanism, at least for smaller input sizes. Among the three, the best social welfare is achieved PS-welfare, since it was designed to extract higher levels of welfare.

In terms of fairness, the optimal mechanism fairs worse in comparison to our mechanisms. From the two versions of PS, PS-welfare is also slightly more fair, which suggests that it is a better choice than its norm-counterpart for this particular problem. Interestingly, RSC outperforms all mechanisms in terms of fairness by a big margin. In fact, for some input sizes, the proportion of envious agents is less than 40% whereas for the optimal mechanism it is close to 80%. This suggests that among the mechanisms we consider, RSC is the one that achieves the best fairness/efficiency guarantees.

The final result seems to be in contrast to the theoretical superiority of Probabilistic Serial over Random Serial Dictatorship in terms of fairness but it can be attributed to two factors: the inputs are not worst-case inputs and more importantly, it seems that the assignment produced by RSC is "closer" to the one outputted by Random Serial dictatorship, when compared to the outputs of Probabilistic Serial and its 3-restricted counterparts.

6 Conclusion

We considered the problem of random assignments in barter exchanges under the additional constraint on the cycle-length of the decomposition. We proposed two new mechanisms for the problem and a general method for designing mechanisms that produce assignments with small cycles. We evaluated our mechanisms on instances of the kidney exchange problem and found that they are better in terms of fairness and not much worse in terms of efficiency, when compared with the optimal exchange. An interesting future direction is to consider other notions of fairness and design different mechanisms for achieving better tradeoffs between fairness and efficiency. The 2-cycle Probabilistic Serial mechanism of Balbuzanov [4] seems like an obvious choice, given that it is designed to produce 3-restricted assignments.

References

1. Abdulkadiroğlu, A., Sönmez, T.: Random serial dictatorship and the core from random endowments in house allocation problems. Econometrica pp. 689–701 (1998)
2. Abraham, D.J., Blum, A., Sandholm, T.: Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In: Proceedings of the 8th ACM conference on Electronic commerce. pp. 295–304. ACM (2007)
3. Ashlagi, I., Roth, A.: Individual rationality and participation in large scale, multi-hospital kidney exchange. In: Proceedings of the 12th ACM conference on Electronic commerce. pp. 321–322. ACM (2011)
4. Balbuzanov, I.: Short trading cycles: Kidney exchange with strict ordinal preferences (2014)
5. Bogomolnaia, A., Moulin, H.: A new solution to the random assignment problem. Journal of Economic Theory 100(2), 295–328 (2001)
6. Dickerson, J.P., Goldman, J.R., Karp, J., Procaccia, A.D., Sandholm, T.: The computational rise and fall of fairness. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada. pp. 1405–1411 (2014)
7. Dickerson, J.P., Procaccia, A.D., Sandholm, T.: Dynamic matching via weighted myopia with application to kidney exchange. In: AAAI (2012)
8. Dickerson, J.P., Procaccia, A.D., Sandholm, T.: Failure-aware kidney exchange. In: Proceedings of the fourteenth ACM conference on Electronic commerce. pp. 323–340. ACM (2013)
9. Dickerson, J.P., Procaccia, A.D., Sandholm, T.: Price of fairness in kidney exchange. In: International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014. pp. 1013–1020 (2014)

10. Goel, A., Kapralov, M., Khanna, S.: Perfect matchings in $o(n \log n)$ time in regular bipartite graphs. SIAM Journal on Computing 42(3), 1392–1404 (2013)
11. Katta, A.K., Sethuraman, J.: A solution to the random assignment problem on the full preference domain. Journal of Economic theory 131(1), 231–250 (2006)
12. Li, J., Liu, Y., Huang, L., Tang, P.: Egalitarian pairwise kidney exchange: fast algorithms vialinear programming and parametric flow. In: International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014. pp. 445–452 (2014)
13. Roth, A.E., Sönmez, T., Ünver, M.: Kidney exchange. The Quarterly Journal of Economics 119(2), 457–488 (2004)
14. Roth, A.E., Sönmez, T., Ünver, M.U.: Pairwise kidney exchange. Journal of Economic theory 125(2), 151–188 (2005)
15. Saban, D., Sethuraman, J.: The complexity of computing the random priority allocation matrix. In: WINE. p. 421 (2013)