http://pubsonline.informs.org/journal/mnsc/

MANAGEMENT SCIENCE
Vol. 65, No. 4, April 2019, pp. 1768–1791
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Failure-Aware Kidney Exchange

John P. Dickerson,[a] Ariel D. Procaccia,[b] Tuomas Sandholm[b]

[a] Department of Computer Science, University of Maryland, College Park, Maryland 20742; [b] Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

Contact: john@cs.umd.edu, http://orcid.org/0000-0003-2231-680X (JPD); arielpro@cs.cmu.edu (ADP); sandholm@cs.cmu.edu (TS)

Received: August 11, 2015
Revised: December 11, 2016
Accepted: August 16, 2017
Published Online in Articles in Advance: July 12, 2018

https://doi.org/10.1287/mnsc.2018.3026

Abstract. Algorithmic matches in fielded kidney exchanges do not typically result in an actual transplant. We address the problem of cycles and chains in proposed matches failing *after* the matching algorithm has committed to them. We show that failure-aware kidney exchange can significantly increase the expected number of lives saved (i) in theory, on random graph models; (ii) on real data from kidney exchange match runs between 2010 and 2014; and (iii) on synthetic data generated via a model of dynamic kidney exchange. This gain is robust to uncertainty over the true underlying failure rate. We design a branch-and-price–based optimal clearing algorithm specifically for the probabilistic exchange clearing problem and show that this new solver scales well on large simulated data, unlike prior clearing algorithms. Finally, we show that failure-aware matching can increase overall system efficiency and simultaneously increase the expected number of transplants to *highly sensitized* patients, in both static and dynamic models.

History: Accepted by Yinyu Ye, optimization.

Funding: This work was supported by the the National Science Foundation [Grants IIS-0905390, IIS-0964579, IIS-1320620, IIS-1546752, IIS-1617590, IIS-1718457, CCF-1101668, and CCF-1733556], the Army Research Office [Grant W911NF-17-1-0082], and a National Defense Science and Engineering Graduate fellowship awarded through the Army Research Office, and used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Foundation [Grant OCI-1053575]; specifically, it used the Blacklight supercomputer at the Pittsburgh Supercomputing Center (PSC).

Keywords: kidney exchange • stochastic matching • stochastic set packing • maximum expected weight cycle cover • random graphs

1. Introduction

Kidney exchange is a recent innovation that allows patients who suffer from terminal kidney failure, and have been fortunate enough to find a willing but incompatible kidney donor, to swap donors. Indeed, it may be the case that two donor-patient pairs are incompatible, but the first donor is compatible with the second patient, and the second donor is compatible with the first patient; in this case, a life-saving match is possible. As we discuss below, sequences of swaps can even take the form of long cycles or chains.

The need for successful kidney exchanges is acute because demand for kidneys is far greater than supply. Although receiving a deceased-donor kidney is a possibility, 36,158 people joined the national waiting list in 2014, while only 16,893 left it that year because they received a kidney. With a median waiting time ranging from two to five years depending on blood type, for some patients kidney exchange is the only viable option.

In this paper,[1] we share learnings from our involvement in designing and running the kidney exchange that was set up by the United Network for Organ Sharing (UNOS). The exchange went live in October 2010, conducting monthly match runs. Since then, the exchange has grown to encompass 143 transplant centers (about 61% of the transplant centers that perform living-donor transplantation in the United States) and now conducts biweekly match runs. Based on this experience, we propose a significantly different approach as a solution to one of the main problems kidney exchanges face today: "last-minute" failures.[2] We mean failures before the transplant surgery takes place, not failures during or after it. Amazingly, most planned matches fail to go to transplant! In the case of the UNOS exchange, 93% of matches fail (Kidney Paired Donation Work Group 2012), and most matches fail at other kidney exchanges as well (e.g., Ashlagi et al. 2011, Leishman et al. 2013, Bray et al. 2015). There are a myriad of reasons for these failures, as we will detail in this paper.

To address such failures, we propose to move away from the deterministic clearing model used by kidney exchanges today into a probabilistic model where the input includes failure probabilities on possible planned transplants, and the output is a transplant plan with maximum *expected* value. The probabilistic approach has recently also been suggested by others (e.g., Chen et al. 2012, Li et al. 2011). They used a general-purpose integer program solver (Gurobi) to

solve their optimization models. We show that general-purpose solvers do not scale to today's real kidney exchange sizes. Then we develop a custom branch-and-price–based (see Barnhart et al. 1998) integer program solver specifically for the probabilistic clearing problem, and show that it scales dramatically better. We also present new theoretical and experimental results that show that the probabilistic approach yields significantly better matching than the current deterministic approach. We conduct experiments both in the static and dynamic setting with (to our knowledge) the most realistic instance generators—one from Saidman et al. (2006) and one that we created that uses real data from all of the UNOS match runs conducted so far—and simulator to date. Perhaps of greatest practical interest to policymakers, we show that, even when higher edge failure rates are correlated with other marginalizing characteristics of a vertex, failure-aware matching can simultaneously increase both the overall number of transplants and the number of transplants to these marginalized patients—in both the static and dynamic settings, on real and simulated data.

1.1. Related Work

The idea of kidney exchange was introduced by Rapaport (1986), and ethical considerations were discussed by Ross et al. (1997). Korea fielded the first kidney exchange program in the 1990s (Park et al. 1999); the first organized exchange in the United States, the New England Paired Kidney Exchange (NEPKE), began in 2003 (see Roth et al. 2004, 2005a, 2007). The topic has attracted—and fielded exchanges have benefited from the work of—researchers from nonmedical fields such as economics (e.g., Roth et al. 2004, 2005a, 2007; Ünver 2010; Yılmaz 2011; Akbarpour et al. 2014; Sönmez and Ünver 2014), operations research (e.g., Biró et al. 2009, Ashlagi et al. 2013, Ashlagi and Roth 2014, Anderson 2014, Glorie et al. 2014, Manlove and O'Malley 2015, Anderson et al. 2015a), and computer science (e.g., Abraham et al. 2007; Awasthi and Sandholm 2009; Toulis and Parkes 2015; Dickerson et al. 2012a, b; Blum et al. 2013; Anshelevich et al. 2013; Dickerson et al. 2014b; Dickerson and Sandholm 2014; Liu et al. 2014; Li et al. 2014). The market for kidneys is constrained by the widespread view (Leider and Roth 2010) that exchanging money for organs is "repugnant" (Roth 2007); thus, in nearly all countries including the United States, it is illegal to buy or sell an organ, which makes deceased- and living-donor donation the only option for procurement of a kidney.

There has been some prior work on the dynamics of kidney exchange, but that work has largely focused on the dynamics driven by pairs and altruists arriving into, and departing from, the exchange rather than on the dynamics driven by failures (Awasthi and Sandholm 2009, Ünver 2010, Dickerson et al. 2012a, Ashlagi et al. 2013, Akbarpour et al. 2014, Anderson et al. 2015a). Also, the techniques developed in those prior papers are completely different from the ones we develop here (and deal with less general models than those that we consider).

Analysis of kidney exchange using random graph models is nowadays the typical method for providing theoretical guidance. Indeed, some of the dynamic kidney exchange papers cited above work in dynamic random graph models (Ünver 2010, Ashlagi et al. 2013, Akbarpour et al. 2014, Anderson et al. 2015a). We work with the model of Ashlagi et al. (2012); related random models include those of Ashlagi and Roth (2014) and Toulis and Parkes (2015).

The work on the query-commit problem by Molinaro and Ravi (2011) is motivated by the same issue as our paper. They study bipartite matching (which equates to clearing with 2-cycles only and no chains in the kidney exchange context) where edges can be tested to see whether they fail. In the query-commit model, if an edge does not fail, it has to be matched. Under certain additional theoretical assumptions, they prove near-optimality of their proposed testing policies. Goel and Tripathi (2012) also study matching with 2-cycles; they provide a greedy testing algorithm for the query-commit problem with an approximation ratio of 0.56 and show that no algorithm can obtain a better ratio for that problem than 0.7916.

Given the ability to perform prematch tests for the existence of up to *two* incident edges per patient–donor pair (instead of the current standard of one), Blum et al. (2013) study the problem of selecting a subset of edges such that expected cardinality of the resulting matching is maximized. They work with only unweighted 2-cycles and no chains, and provide a polynomial time algorithm that almost surely maximizes (up to lower-order terms) the expected number of swaps in that model.

Subsequent to the initial posting of the present paper, Anderson (2014), Anderson et al. (2015b), Blum et al. (2015), Dickerson and Sandholm (2015), Assadi et al. (2016), and Dickerson et al. (2016) each look at match failures in a variety of models. Anderson (2014) and Anderson et al. (2015b) first present a scalable deterministic kidney exchange algorithm that is not amenable to failure-aware matching, but also study the problem through the lens of two-stage stochastic optimization. Dickerson et al. (2016) build on that work and give the first compact formulation—that is, a model that is of size polynomial in the size of the input—of kidney exchange; their model is amenable only to failure-aware matching for *constant* failure rates, unlike the present work. With similar motivation, Blum et al. (2015) and Assadi et al. (2016) continue the work of Blum et al. (2013) and look at nonadaptive and

adaptive policies for selecting edges to test before performing a final maximum matching; however, their analysis caters to cycles only. Dickerson and Sandholm (2015) build on techniques from the present paper and from Dickerson et al. (2012a, 2014b) to learn using data how to match in a realistic model of dynamic kidney exchange; their framework also uses machine learning and data to instantiate human experts' high-level goals into a concrete objective function for optimization. Glorie et al. (2014) present a branch-and-price solver that, under certain assumptions, is able to solve the pricing problem for new cycles and chains in polynomial time. They do not explicitly consider postalgorithmic match failure. Also, the assumptions required for their polynomial-time pricing problem solution break under the addition of failure probabilities to edges in chains, although they do hold for problems with cycles only.

2. Modeling Expected Utility: Considering Cycle and Chain Failure

In this section, we augment the standard model of kidney exchange to include the probability of edge, cycle, and chain failure. We formalize the *expected utility* of an edge, cycle, and chain, which represents the expected number of actual transplants (not just potential transplants). This is used to define the expected utility of an overall matching, which more accurately reflects its value relative to other matchings.

2.1. The Basic Graph Model for Kidney Exchange

The standard model for kidney exchange encodes an n-patient kidney exchange—and almost any n-participant barter exchange—as a directed *compatibility graph* $G(n)$ by constructing one vertex for each patient–donor pair. An edge e from v_i to v_j is added if the patient in v_j wants, and could potentially receive, the donor kidney (or item, in general) of v_i; in this case, we say the patient of v_j is *compatible* with the donor from v_i. A donor is willing to give her kidney if and only if the patient in her vertex v_i receives a kidney.

The weight w_e of an edge e represents the utility to v_j of obtaining v_i's donor kidney (or item). A cycle c in the graph G represents a possible kidney swap, with each vertex in the cycle obtaining the kidney of the previous vertex. If c includes k patient–donor pairs, we refer to it as a *k-cycle*. In kidney exchange, typically cycles of length at most some small constant L are allowed—all transplants in a cycle must be performed simultaneously so that no donor backs out after his patient has received a kidney but before he has donated his kidney. In most fielded kidney exchanges, including the UNOS kidney exchange, $L = 3$ (i.e., only 2- and 3-cycles are allowed).

Currently, fielded kidney exchanges gain great utility through the use of *chains* (see, e.g., Roth et al. 2006, Montgomery et al. 2006, Rees et al. 2009, Gentry et al. 2009, Ashlagi et al. 2011, Gentry and Segev 2011, Dickerson et al. 2012b, Ashlagi et al. 2012). Chains start with an altruistic donor donating his kidney to a candidate, whose paired donor donates her kidney to another candidate, and so on. Chains can be (and typically are) longer than cycles in practice because it is not necessary to carry out all of the transplants in a chain simultaneously. Of course, there is a chance that a bridge donor backs out of his/her commitment to donate. In that unfortunate event, which has happened a couple of times in the United States, the chain does not continue. Cycles cannot be executed piecemeal because if someone backs out of a cycle, then some pair has lost a kidney (i.e., their "bargaining chip"). In contrast, if someone backs out of a chain, no pair has lost their bargaining chip (although of course it is a shame if some chain does not continue forever).

A *matching* M is a collection of disjoint cycles and chains in the graph G. The cycles and chains must be disjoint because no donor can give more than one of her kidneys.

2.2. Including Failure Probability in the Model

In the basic kidney exchange model, the weight w_c of a cycle or chain c is the sum of its edge weights, and the weight of a matching M is the sum of the weights of the cycles and chains in the matching. The *clearing problem* is then to find a maximum (weighted) matching M. In reality, not all of the recommended matches proceed to transplantation, because of varying levels of sensitization between candidates and donors in the pool (represented by a scalar factor called CPRA), illness, uncertainty in medical knowledge, or logistical problems. As such, the weight of a cycle or chain as the sum of its constituent parts does not fully characterize its true worth.

Associate with each edge $e = (v_i, v_j)$ in the graph G a value $q_e \in [0, 1]$ representing the probability that, if algorithmically matched, the patient of v_j would successfully receive a kidney from v_i's donor. We will refer to q_e as the *success probability* of the edge, and $1 - q_e$ as the *failure probability* of the edge. Using this notion of failure probability, we can define the expected (failure-aware) utility of chains and cycles.

2.2.1. Expected Utility of a Cycle.

For any transplant in a k-cycle to execute, each of the k transplants in that cycle must execute. Put another way, if even one algorithmically matched transplant fails, the entire cycle fails. Thus, for a k-cycle c, define the *expected utility* $u(c)$ of that cycle to be

$$u(c) = \left[\sum_{e \in c} w_e\right] \cdot \left[\prod_{e \in c} q_e\right].$$

That is, the utility of a cycle is the product of the sum of its constituent weights and the probability of the cycle

executing. The simplicity of this calculation is driven by the required atomicity of cycle execution—a property that is not present when considering chains.

2.2.2. Expected Utility of a Chain. While cycles must execute entirely or not at all, chains can execute incrementally. For example, a 3-chain $c = (a, v_1, v_2, v_3)$ starting at altruist a might result in one of four numbers of transplants:

- No transplants, if the edge (a, v_1) fails.
- A single transplant, if (a, v_1) succeeds but (v_1, v_2) fails.
- Two transplants, if (a, v_1) and (v_1, v_2) succeed but (v_2, v_3) fails.
- Three transplants, if all edges in the chain represent successful transplants. (In this case, the donor at v_3 typically donates to the deceased donor waiting list, or stays in the pool as a bridge donor. Whether or not this additional transplant is included in the optimization process is decided by each individual kidney exchange program.)

In general, for a k-chain $c = (v_0, v_1, \ldots, v_k)$, where v_0 is an altruist, there are k possible matches (and the final match to, for example, a deceased donor waiting list candidate). Let q_i be the probability of edge $e_i = (v_i, v_i + 1)$ leading to a successful transplant. Here, we assume $w_e = 1$ for ease of exposition; in Section 5, we show that relaxing this assumption does not complicate matters.

Then, the expected utility $u(c)$ of a k-chain c is

$$u(c) = \left[\sum_{i=1}^{k-1}(1-q_i)i\prod_{j=0}^{i-1}q_j\right] + \left[k\prod_{i=0}^{k-1}q_i\right].$$

The first portion above calculates the sum of expected utilities for the chain executing exactly $i = \{1, \ldots, k-1\}$ steps and then failing on the $(i+1)$th step. The second portion is the utility gained if the chain executes completely.

2.2.3. Expected Utility of a Matching. The value of an individual cycle or chain hinges on the interdependencies between each specific patient and potential donor, as was formalized above. However, two cycles or chains in the same matching M fail or succeed independently. Thus, define the expected utility of a matching M to be

$$u(M) = \sum_{c \in M} u(c).$$

That is, the expected number of transplants resulting from a matching M is the sum of the expected number of transplants from each of its constituent cycles and chains.

For a (possibly weighted) compatibility graph $G = (V, E)$, let $\mathcal{M}$ represent the set of all legal matchings induced by G. Then, given success probabilities q_e, $\forall\, e \in E$, the *failure-aware clearing problem* is to find M^* such that

$$M^* = \arg\max_{M \in \mathcal{M}} u(M).$$

The distinction between M^* and any maximum (non-failure-aware) weighted matching M' is important, as we show in the rest of this paper—theoretically in Section 3, on real data from the fielded UNOS kidney exchange in Section 4, and on simulated data in Section 7.

3. Maximum Cardinality Matching Is Far from Maximizing the Expected Number of Transplants

In this section, we prove a result regarding the (in)efficacy of maximum cardinality matching in kidney exchange, when the probability of a match failure is taken into account. We show that in pools containing equally sensitized patient–donor pairs (and not necessarily equally sensitized altruistic donors), with high probability there exists a "failure-aware" matching that has linearly higher expected utility than all maximum cardinality matchings. This theoretical result motivates the rest of the paper; since current fielded kidney exchanges perform maximum cardinality or maximum weighted matching, many potential transplants may be left on the table as a consequence of not considering match failures.

3.1. Random Graph Model of Sensitization

We work with (a special case of) the model of Ashlagi et al. (2012, section 4.2), which is an adaptation of the standard *theoretical* kidney exchange model to pools with highly and nonhighly sensitized patient–donor pairs.

The model works with random compatibility graphs with $n + t(n)$ vertices, pertaining to n incompatible patient–donor pairs (denoted by the set P), and $t(n)$ altruistic donors (denoted by the set A), respectively. Edges between vertices represent not just blood-type compatibility, but also immunological compatibility—the *sensitization* of the patient. Given a blood-type–compatible donor, let p denote the probability that an edge exists between a patient and that donor.

Given uniform sensitization p for each of the n patients in the pool, random graphs from this model are equivalent to those of Erdős and Rényi (1960) with parameters n and p. In Section 3.2, we use techniques from random graph analysis to prove that maximum cardinality matching in highly sensitized pools (with altruists) does not optimize for expected number of actual transplants.

3.2. Maximum Cardinality Matching in Highly Sensitized Pools

Let $G(n, t(n), p)$ be a random graph with n patient–donor pairs, $t(n)$ altruistic donors, and probability

$p = \Theta(1/n)$ of incoming edges. Such a p represents highly sensitized patients. Let q be the probability of transplant success that we introduced, such that q is constant for each edge e. Note that for a chain of length k, the probability that $t < k$ matches execute is $q^t(1-q)$, and the probability that k matches execute is q^k. There is no chain cap (although we could impose one, which depends on q). Given a matching M, let $u_q(M)$ be its expected utility in our model—i.e., expected number of successful transplants. Denote the set of altruistic donors by A, and denote the vertex pairs by P.

The proof of the following theorem builds on techniques used in the proof of theorem 5.4 of Ashlagi et al. (2012), but also requires several new ideas.

Theorem 1. *For every constants $q \in (0,1)$ and $\alpha, \beta > 0$, given a large random graph $G(n, \alpha n, \beta/n)$, with high probability there exists a matching M' such that for every maximum cardinality matching M,*

$$u_q(M') \geq u_q(M) + \Omega(n).$$

Proof. We consider subgraphs that we call *Y-gadgets*, with the following structure. A Y-gadget contains a path $(u, v_1, \ldots, v_k)$ such that $u \in A$ and $v_i \in P$ for $i = 1, \ldots, k$ for a large enough constant k, to be chosen later. Furthermore, there is another altruistic donor $u' \in A$ with two outgoing edges, (u', v_3) and (u', v') for some $v' \in P$. Finally, the edges described above are the *only* edges incident on the vertices of the Y-gadget. See Figure 1(a) for an illustration.

We first claim that it is sufficient to demonstrate that with high probability the graph $G(n, \alpha n, \beta/n)$ contains cn Y-gadgets, for some constant $c > 0$. Indeed, because each Y-gadget is disconnected from the rest of the graph, a maximum cardinality matching M must match all of the vertices of the Y-gadget, via a k-chain and a 1-chain. Let M_Y be the restriction of M to the Y-gadget (see Figure 1(b)). It holds that

$$u_q(M_Y) = (1-q)\sum_{i=1}^{k-1} iq^i + kq^k + q.$$

We next construct a matching M'_Y for the Y-gadget, via two chains: (u, v_1, v_2) and $(u', v_3, \ldots, v_k)$—i.e., vertex v' remains unmatched (see Figure 1(c)). We obtain

$$u_q(M'_Y) = (1-q)\sum_{i=1}^{k-3} iq^i + (k-2)q^{k-2} + q(1-q) + 2q^2.$$

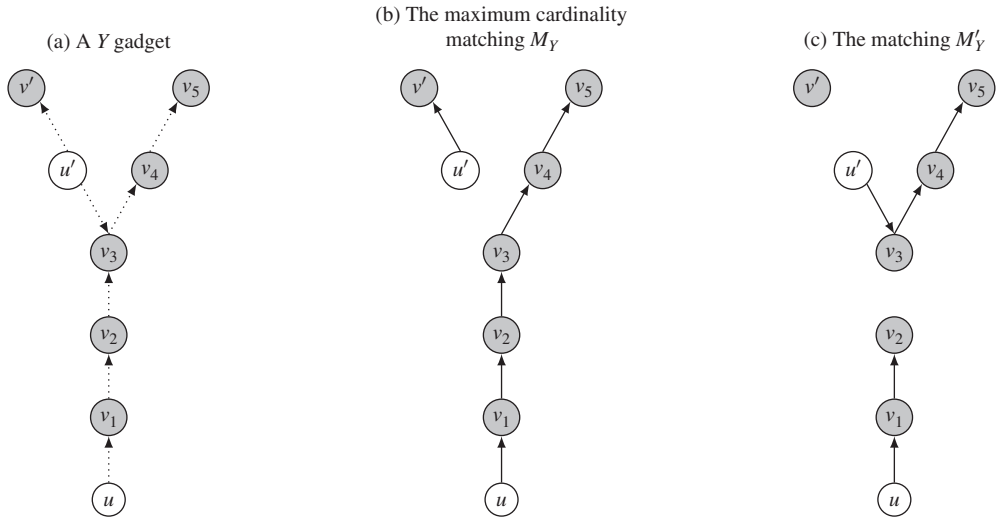
Therefore,

$$u_q(M'_Y) - u_q(M_Y) = q^2 + (k-2)q^{k-1} - (k-1)(1-q)q^{k-1} - kq^k > q^2 - (k+1)q^{k-1}. \quad (1)$$

Clearly if k is a sufficiently large constant, $q^2/2 > (k+1)q^{k-1}$, and hence the right-hand side of Equation (1) is at least $q^2/2$, which is a constant. By applying this argument to each of the cn Y-gadgets, we obtain a matching M' such that $u_q(M') - u_q(M) > (q^2/2)cn = \Omega(n)$.

It remains to establish the existence of $\Omega(n)$ Y-gadgets. Consider a random undirected graph with $n + \alpha n$ vertices. The edge probabilities are $p = 2(\beta/n)(1 - \beta/n) + (\beta/n)^2$—i.e., the probability of at least one edge existing between a pair of vertices in P. A standard result on sparse random graphs (see, e.g., Janson et al. 2011) states that for every graph X of constant size, with high probability we can find $\Omega(n)$ subgraphs that

Figure 1. Illustration of a Y-Gadget with $k = 5$



Notes. The vertices of A are white and the vertices of P are gray. Clearly $|M_Y| > |M'_Y|$, but (using Equation (1)) $u_q(M'_Y) - u_q(M_Y) > q^2 - 6q^4$; this difference is positive for $q < 0.41$, which is a realistic value.

are *isomorphic* to X and *isolated* from the rest of the graph. In particular, with high probability our random graph has $\Omega(n)$ subgraphs that are isomorphic to the undirected, unlabeled version of a Y-gadget.

There are two independent issues we need to address. First, these subgraphs are unlabeled—i.e., we do not know which vertices are in A and which are in P. Second, the graph is undirected, and may have some illegal edges between pairs of vertices in A. We presently address the first issue. We randomly label the vertices as A or P, keeping in mind that ultimately it must hold that $|P| = n$ and $|A| = \alpha n$. Assume without loss of generality that $\alpha \leq 1$. Consider an arbitrary vertex in one of the special subgraphs. This vertex is in P with probability $1/(1+\alpha)$, and in A with probability $\alpha/(1+\alpha)$. The label of the second vertex depends on the first. For example, if the first is in P, then the probabilities are $(1-1/n)/(1-1/n+\alpha)$ for P and $\alpha/(1-1/n+\alpha)$ for A.

We sequentially label the vertices of $\min\{cn, (\alpha n)/10k)\}$ gadgets, where cn is the number of Y-gadgets, taking into account the previous labels we observed. (Note that we are labeling a linear number of Y-gadgets, since k is constant.) Because we observed far fewer than $\alpha n/2$ labels, in each trial the probability of each of the two labels, conditioned on the previous labels, is at least $(\alpha/2)/(1+\alpha/2)$, which is a constant. This lower bound allows us to treat the labels as independent Bernoulli trials. Thus, the probability that a gadget has exactly the right labels (two A labels in the correct places, and P labels everywhere else) is at least $r = ((\alpha/2)/(1+\alpha/2))^{k+3}$, which is a constant. The expected number of correctly labeled gadgets is therefore at least $r \cdot \min\{cn, \alpha n/(10k)\}$—i.e., $\Omega(n)$. Using Chernoff's inequality, with high probability we can find $\Omega(n)$ correctly labeled gadgets.

We next address the second issue: the directions of the edges. For each of the $\Omega(n)$ correctly labeled gadgets, each undirected edge corresponds to a directed edge in one of the two direction with probability

$$\frac{(\beta/n)(1-\beta/n)}{2(\beta/n)(1-\beta/n)+(\beta/n)^2} \approx \frac{1}{2},$$

and corresponds to edges in both directions with the complement (small) probability. The probability that each edge in a Y-gadget corresponds to a single edge in the correct direction is therefore constant, and using similar arguments as above, with high probability a constant fraction of the correctly labeled gadgets will have correctly oriented edges.

Finally, note that the labels of the vertices and the directions of the edges in each of the initially unlabeled, undirected Y-gadgets are independently assigned. Given one of these initial (linearly many) Y-gadgets, we have shown that the probability of that Y-gadget being labeled correctly is a constant. Similarly, we have shown that the probability of that Y-gadget having all edges in exactly the right orientations is a constant. Thus, since these events are independent, the probability that both events occur is a constant, and we have a constant fraction of the linearly many initial Y-gadgets with the correct orientation of edges and labels of nodes.

A final issue remaining to be addressed is that there are no edges between pairs of vertices in A, and the probability of edges (u, v) where $u \in A$ and $v \in P$ is smaller than p. We first note that, since we are looking at a denser graph, isolation is harder to achieve. Moreover, since a Y-gadget has no adjacent vertices in A, we discard any such Y-gadgets, so the increased probability of edges between such pairs does not help us. Finally, for pairs (u, v) where $u \in A$ and $v \in P$, the probability of an edge (u, v) existing is *equal* to the probability of an edge existing in the undirected graph *and* the edge being in the right direction; Y-gadgets where the edge is in the wrong direction are discarded. □

Importantly, while the proof of Theorem 1 only explicitly discusses chains (in the construction of the Y-gadgets), the optimal matching also contains cycles—they are just not the driving force behind this result. In the next section, we provide experimental validation of this theoretical result using real data from the UNOS nationwide kidney exchange, which we help run.

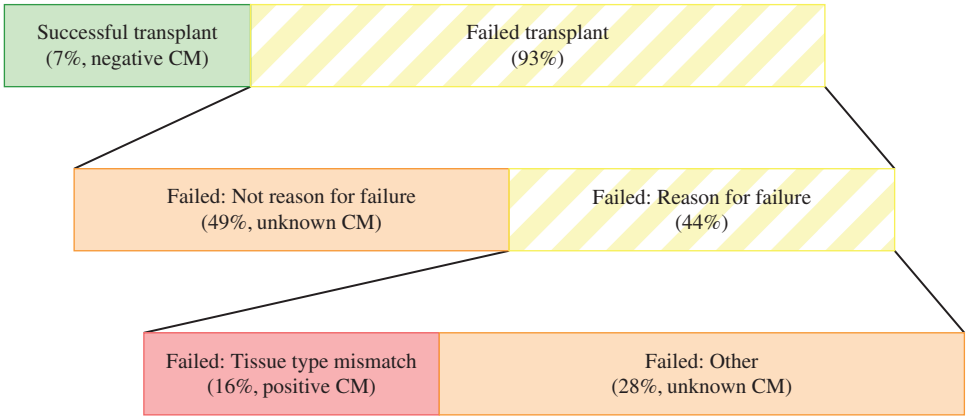
4. Experiences from, and Experiments on, the UNOS Kidney Exchange

Over the past decade, fielded kidney exchanges have begun appearing in the United States. One of the largest, run by the United Network for Organ Sharing (UNOS), performed its first match run in October of 2010. As of July 2015, it matches on a biweekly basis, and interacts nationwide with 143 transplant centers. We maintain the optimization code for match runs in the UNOS kidney exchange program, and interact frequently with the medical, logistical, and support staff for the program. In this section, we present experimental results comparing failure-aware and deterministic matching on real data from this exchange, using multiple estimated distributions over edge failure probabilities. We also perform a sensitivity analysis on the gains realized by the algorithm when there is uncertainty over the true underlying failure probabilities, as is the case in reality.

4.1. Estimating Edge Failure Probabilities

The UNOS U.S.-wide kidney exchange computes a maximum weighted matching at each clearing. The function used to assign weights to edges was determined by a committee of medical professionals, and

Figure 2. (Color online) Determining the Probability of a Match Failing Is Difficult Because Many Potential Patient–donor Pairs Are Not Crossmatched



Note. Of the aggregate UNOS data, we are only sure that the 7% who successfully received a transplant and the 16% who explicitly failed because of a positive crossmatch were tested.

takes into account such factors as donor and patient location, health, and CPRA score. We have access to these data and use them in our experiments.

However, medical knowledge is incomplete; as such, we cannot determine the exact probability q that a potential transplant will succeed. For our experiments, we use multiple distributions of edge failure probabilities.

First, we draw from all of the data from the match runs conducted in the UNOS exchange to date. Figure 2 displays success and failure results for recommended matches from the UNOS kidney exchange for matches between October 27, 2010, and November 12, 2012.[3] Approximately 7% of matches resulted in a transplant, while approximately 93% failed. Of the 93% that failed:

- 49% were not the reason for failure. The cycle or chain in which the potential transplant was involved failed entirely (in the case of cycles) or before the patient's turn (in the case of chains).
- 44% were the reason for failure.

—36% of these (about 16% of the total) failed because of a *positive crossmatch*, signifying blood-type incompatibility (beyond the ABO model).

—64% failed for a variety of other reasons, as discussed below.

Triggering a cycle or chain failure can occur for a variety of reasons, including

- receiving a transplant from the deceased donor waiting list;
- receiving a transplant from another exchange;
- patient or donor becoming too ill for surgery or expiring;
- an altruistic patient "running out of patience" and donating elsewhere, or not at all;
- a donor in a chain reneging (i.e., backing out after his patient received a kidney);
- pregnancy or sickness changing a patient or donor's antigen incompatibilities.

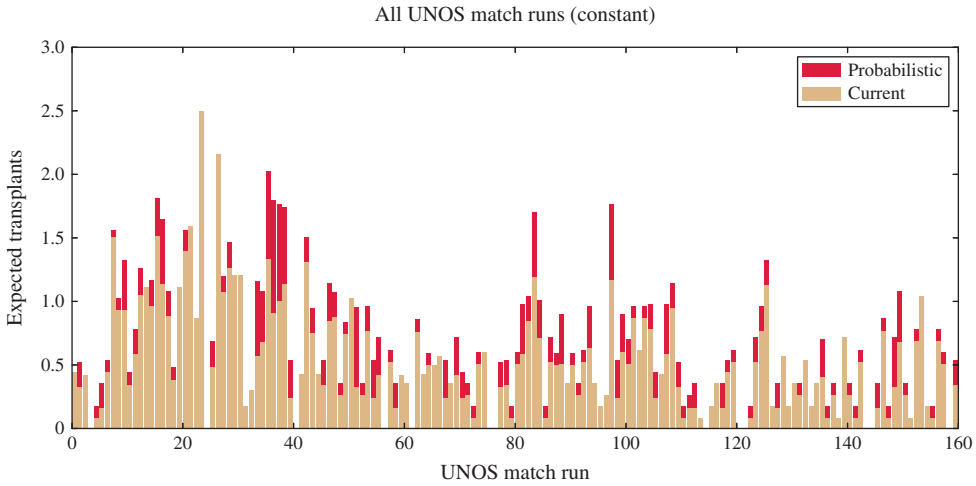
In these cases, a patient and potential donor may or may not have received a crossmatch test. In fact, the only sureties regarding crossmatches to be derived from the data above are that 7% crossmatched negative (those who received transplants) and 16% crossmatched positive. Thus, roughly $7/(16+7) \approx 30\%$ of these crossmatches came back negative. We use this value for our first set of simulations, setting the probability of a crossmatch failing to be a constant 70%. This 70% expected failure is optimistic (i.e., too low) in that it ignores the myriad other reasons for match failures. UNOS currently performs batch myopic matches, so—for these simulations—we only simulated crossmatch failures. We take additional failure reasons into account in Sections 7 and 8.

Second, in the UNOS exchange and in others (see., e.g., Ashlagi et al. 2012), patients tend to have either very high or very low sensitization—i.e., there is a very low or very high probability that their blood will pass a crossmatch test with a random organ. For *highly sensitized* patients, finding a kidney is very difficult. Drawing from this and the 70% failure rate derived above, our second set of experiments samples randomly from a bimodal distribution: 25% of edges have a low failure rate $(1-q_L) \in U[0.0, 0.2]$, while 75% have a high failure rate $(1-q_H) \in U[0.8, 1.0]$, such that the overall expected failure rate is 70%. Third, we systematically vary the variance of the underlying failure probability distribution and explore its effect on the behavior of both matching methods.

4.2. UNOS Results: Failure-Aware Matching Is Better in Practice

We now simulate probabilistic matching on real data from UNOS, using that exchange's cycle length cap

Figure 3. (Color online) Comparison of the Expected Number of Transplants Resulting from the Maximum Weighted Matching and Failure-Aware Weighted Matching Methods, on 161 UNOS Match Runs Between October 2010 and November 2014, with a Constant Edge Success Probability



of 3. We performed simulations using both the constant 70% failure rate and the bimodal failure rate. On the former, we can compute an exact expected value for the failure-aware matching on each real UNOS matching. On the latter, we simulated failure probabilities at least 100 times for each UNOS match run.

Figures 3 and 4 show that, in both cases, taking failure probabilities into account results in significantly more expected transplants. In the constant probability case, failure-aware matching yields many more matches than (or in some cases the same number as) the status quo of maximum weighted matching. (In cases where the expected utility of both matching methods was equal, the matchings with equivalent compositions (i.e., same number of 2-cycles, 3-cycles, and k-chains) were returned by both solvers.) The failure-aware matching performed better when the maximum weighted matching included long chains, a frequent phenomenon in the UNOS pool (and other fielded exchange pools in the United States and abroad), as discussed by Dickerson et al. (2012b), Ashlagi et al. (2012), and Glorie et al. (2014).

In the bimodal case, failure-aware matching shines, often beating the maximum cardinality matching by a factor between 2 and 5, and again never doing worse in expectation. Here, the failure-aware matching algorithm is able to pick cycles and chains that contain edges with very high probabilities of success over those with very low probabilities of success.

Table 1 gives aggregate match data for both the current UNOS solver and our proposed method on both the constant and bimodal underlying failure rate

Figure 4. (Color online) Comparison of the Expected Number of Transplants Resulting from the Maximum Weighted Matching and Failure-Aware Weighted Matching Methods, on 161 UNOS Match Runs Between October 2010 and November 2014, with Bimodal Edge Success Probabilities (Some Very High, Some Very Low)

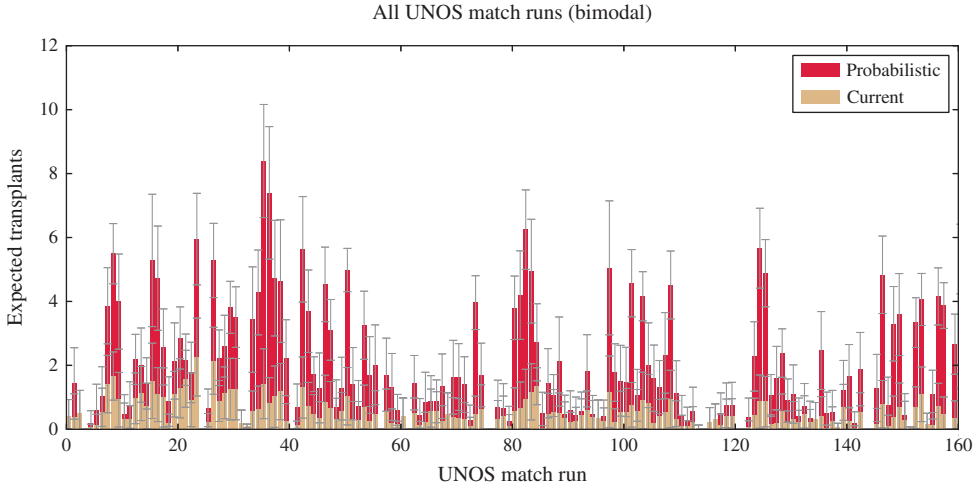


Table 1. Distributional Difference Between Maximum Weighted Matching and Failure-Aware Matching on Real UNOS Data

	Current		Ours		t-test		Wilcoxon signed-rank	
Distribution	Average	Std. dev.	Average	Std. dev.	t-statistic	p-value	Siegel's T	p-value
Constant	0.52	0.43	0.67	0.50	10.95	$<10^{-10}$	0	$<10^{-10}$
Bimodal	0.51	0.43	1.89	1.79	11.85	$<10^{-10}$	0	$<10^{-10}$

probability distributions. Across all UNOS match runs using a constant edge failure probability of 0.7, the failure-aware method results in an expected 0.15 more transplants per match run over the maximum weighted matching solver. Using the bimodal distribution, the failure-aware method returns an expected additional 1.38 transplants per match run. Table 1 gives the results of both a paired t-test and a Wilcoxon signed-rank test (a nonparameteric version of the t-test); we ran both on the expected number of transplants from the 161 paired deterministic and failure-aware optimal matchings for each of the UNOS match runs to test if their population means were different. Clearly, the gains seen under both failure distributions are statistically significant.

4.3. Distributional Diversity Begets Greater Gains

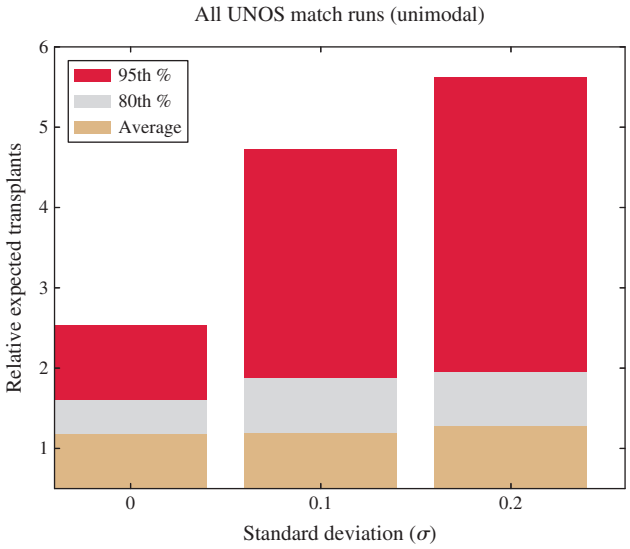
Section 4.2 showed experimentally that (i) a statistically significant gain in expected matches occurs under the consideration of match failure and (ii) a bimodal underlying failure probability distribution resulted in more of a gain than a constant underlying failure probability distribution. We delve deeper into this insight in this section.

We now investigate the effect that higher *variance* in edge failure probabilities has on the overall value of both matching methods. For this section's experiments, we sample from a normal distribution with mean of 0.7 and varying standard deviation. If a sample returns an illegal failure probability p (i.e., $p < 0$ or $p > 1$), we resample from the underlying distribution. In this way, we expand the underlying distribution from a constant 0.7 toward a more uniform randomness.

Figure 5 shows the aggregate number of expected transplants (summed over all UNOS match runs through November 2013) for varying levels of variance σ^2, given a standard deviation of σ, in the underlying distribution from which failure probabilities are sampled. For convenience, we label the constant probability of 0.7 case as "$\sigma = 0.0$." Positive crossmatches are simulated based on an edge's sampled probability of failure.

In the constant probability case, failure-aware matching results in an average expected 18.4% increase in expected transplants. As the standard deviation of the underlying distribution increases, so too does this expected boost: from 18.8% to 28.5%, respectively,

Figure 5. (Color online) Aggregate Additional Transplants Over All UNOS Match Runs Through November 2013, for Edge Failure Probabilities Drawn Randomly from $\mathcal{N}(\mu = 0.7, \sigma \in \{0.1, 0.2\})$



Note. The leftmost point "$\sigma = 0.0$" represents a constant failure rate of 70%.

for $\sigma = 0.10$ and $\sigma = 0.20$, respectively. An increase in variance also results in the maximum cardinality matching method frequently missing the highest utility match by a large margin. For instance, the 80th and 95th percentiles increase from an additional 59.8% and 154.2% in the constant probability case to 94.6% and 462.9% when $\sigma = 0.20$. Higher variance results in more opportunities for the maximum cardinality matching to contain many matches with an extremely low probability of execution (e.g., a 3-cycle with edges that are likely to fail instead of a smaller 2-cycle with more reliable edges).

4.4. Robustness to Uncertainty Over True Edge Failure Probabilities

The previous sections assumed that the true underlying failure rate for each edge was known with certainty—that is, if the optimizer uses a failure rate of 0.41 for an edge e, then in reality that edge fails with probability 0.41. In reality, any available failure probabilities would be noisy estimates of the true underlying failure probabilities. For example, Glorie (2012) estimate the probability of a positive crossmatch as a

Table 2. Expected Gain in Number of Realized Transplants When Optimizing Using a Constant Failure Rate of f', Given a True Underlying Failure Rate of f, Compared to Failure-Aware Matching Using the True Failure Rate

	Estimated failure rate f'									
True rate f	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.0	—	0.00	0.00	0.00	−1.14	−1.14	−1.15	−1.15	−1.15	−1.15
0.1	0.00	—	0.01	0.01	−0.62	−0.63	−0.63	−0.63	−0.63	−0.63
0.2	−0.03	0.00	—	0.00	−0.31	−0.30	−0.30	−0.31	−0.30	−0.30
0.3	−0.02	0.01	0.01	—	−0.07	−0.07	−0.07	−0.08	−0.07	−0.07
0.4	−0.12	−0.08	−0.07	−0.08	—	0.00	0.00	0.00	0.00	0.00
0.5	−0.17	−0.14	−0.13	−0.14	0.00	—	0.00	0.00	0.00	0.00
0.6	−0.16	−0.14	−0.14	−0.14	0.00	0.00	—	0.00	0.00	0.00
0.7	−0.14	−0.13	−0.13	−0.13	0.00	0.00	0.00	—	0.00	0.00
0.8	−0.09	−0.07	−0.07	−0.07	0.00	0.00	0.00	0.00	—	0.00
0.9	−0.02	−0.02	−0.02	−0.02	0.00	0.00	0.00	0.00	0.00	—

function of a patient's CPRA, and fit a probit model to predict the crossmatch result. They are able to find a noisy estimate of the underlying failure rate. Using this motivation (that, in reality, it is possible to find a reasonable but imperfect estimate of the true underlying failure rate), we now perform a sensitivity analysis to test our method's robustness to uncertainty over the underlying true edge failure rate.

In the following experiments, we vary two failure rates $f, f' \in \{0.0, 0.1, \ldots, 0.9\}$. We use f to refer to the true (unknown) underlying failure rate for an edge; similarly, we use f' to refer to the estimated failure rate that is given to the optimizer. For example, if $f' = 0.41$ but $f = 0.2$, then the optimizer uses the failure rate 0.41 to produce a failure-aware matching, but each edge will fail (i.i.d.) in reality with probability 0.2. We then measure the number of realized transplants (so, transplants that were achieved after the true failure rate f is applied) and compare against one of two baselines. Experiments are performed on the real UNOS match runs as follows. For each match run, for each true failure rate f, for each edge, simulate if that edge exists or does not exist. Then, for each failure rate f', run the failure-aware matching algorithm using f'. Receive credit only for those cycles and portions of chains that exist given the predetermined failures in accordance with true rate f. This is done 50 times per match run, true rate f, and estimated rate f'.

Table 2 compares the performance of failure-aware matching using f' to that of failure-aware matching using $f' = f$. For example, using a failure rate of $f' = 0.6$ when the true underlying failure rate is $f = 0.3$ results in an expected 0.07 fewer transplants realized than using $f' = f = 0.3$. Intuitively, losses increase as the distance between f and f' increases. Experimentally, there is a large jump in loss when using a failure rate of $f' \geq 0.4$ if $f < 0.4$, and when using $f' < 0.4$ if $f \geq 0.4$. This can be explained analytically as follows. Given a failure probability f, consider the expected utilities of a 2-cycle c_2 and a 3-cycle c_3: $u(c_2) = 2(1-f)^2$ and $u(c_3) = 3(1-f)^3$. When $f < \frac{1}{3}$, $u(c_2) < u(c_3)$, so the optimizer favors 3-cycles over 2-cycles. When $f > \frac{1}{3}$, $u(c_2) > u(c_3)$, so the optimizer favors 2-cycles. So, when f and f' are on different sides of this $\frac{1}{3}$ boundary, the optimizer will either be incorrectly optimistic—resulting in the use of too-risky 3-cycles—or incorrectly pessimistic—resulting in the avoidance of more valuable 3-cycles.

Table 3 uses a different underlying baseline, and compares failure-aware matching using f' to the status quo deterministic matching (which is equivalent to failure-aware matching using $f' = 0$). Here, we see the $\frac{1}{3}$ boundary expressed even more clearly. When the true underlying failure rate f is above that boundary, then failure-aware matching with *any* failure rate outperforms deterministic matching. When both the estimated f' and true f are below the boundary, failure-aware matching outperforms deterministic matching. But, when f is below and f' is above the boundary, the failure-aware matching algorithm is too pessimistic and avoids 3-cycles that, in expectation, are worth using. We note that, in practice, $f \approx 0.7 \gg \frac{1}{3}$; thus, these results support the failure-aware matching method dramatically outperforming the deterministic status quo under even large differences in estimated failure rate f' and true failure rate f.

Next, in Section 5, we construct a solver that is capable of optimally clearing large exchanges—larger, even, than those currently available at UNOS or other fielded kidney exchanges.

5. Building a Scalable Solver to Clear Failure-Aware Exchanges

Current kidney exchange pools are small, containing at most a few hundred patients at a time. For example, so far the UNOS match runs never had pools larger than 258 patients and 277 donors. However, as kidney exchange gains traction, these pools will grow. As discussed by Abraham et al. (2007), the estimated

Table 3. Expected Gain in Number of Realized Transplants When Optimizing Using a Constant Failure Rate of f', Given a True Underlying Failure Rate of f, Compared to the *Deterministic* Status Quo

	Estimated failure rate f'									
True rate f	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.0	—	0.00	0.00	0.00	−1.14	−1.14	−1.15	−1.15	−1.15	−1.15
0.1	—	0.00	0.00	0.01	−0.62	−0.63	−0.63	−0.63	−0.63	−0.63
0.2	—	0.03	0.03	0.02	−0.28	−0.27	−0.28	−0.28	−0.28	−0.28
0.3	—	0.04	0.03	0.02	−0.05	−0.05	−0.05	−0.06	−0.05	−0.05
0.4	—	0.04	0.04	0.03	0.12	0.11	0.11	0.11	0.12	0.11
0.5	—	0.03	0.03	0.03	0.16	0.17	0.16	0.17	0.17	0.16
0.6	—	0.02	0.02	0.02	0.16	0.16	0.16	0.16	0.16	0.16
0.7	—	0.01	0.01	0.01	0.14	0.14	0.14	0.14	0.14	0.14
0.8	—	0.01	0.01	0.01	0.08	0.08	0.09	0.09	0.09	0.09
0.9	—	0.00	0.01	0.00	0.02	0.02	0.02	0.02	0.02	0.02

steady-state size of a U.S. nationwide kidney exchange is 10,000 patients.

Clearing pools of this size is a computational challenge. Abraham et al. (2007) showed that the deterministic clearing problem is NP-hard. Since the deterministic clearing problem is a special case of the failure-aware clearing problem (that is, it is the failure-aware clearing problem with constant success probability $q = 1.0$), it follows that the failure-aware clearing problem is also NP-hard.

Proposition 1. *The failure-aware clearing problem is NP-hard.*

To our knowledge, there is no solver that would scale to the nationwide steady-state size—including the CMU solver used by UNOS. This solver is based on the work of Abraham et al. (2007), with enhancements and generalizations by Dickerson and Sandholm, and uses integer linear programming (IP) with one decision variable x_c for each cycle c no longer than L (in practice, $L = 3$), and constraints that state that accepted cycles are vertex disjoint. For expository ease in the coming sections, Equation (2) presents that IP model, written in the context of $C(L)$, the set of all cycles of length at most L.

$$\max \sum_{c \in C(L)} u(c) x_c \quad \text{s.t.} \sum_{c : v \in c} x_c \leq 1 \quad \forall v \in V \qquad (2)$$

With specialized branch-and-price IP solving techniques, Abraham et al. (2007) were able to solve the (3-cycle, no chains, deterministic) problem at the projected steady-state nationwide scale of 10,000 patients.

In the current UNOS solver, chains are incorporated by adding from the end of each potential chain a "dummy" edge of weight 0 to every vertex that represents an altruist. Chains are generated in the same fashion as cycles, and look identical to cycles to the optimization algorithm—with one caveat. Recall that chains need not be executed atomically, and thus, in practice, the cycle cap of 3 is not applicable to chains. Because of the removal of this length restriction, this approach does not scale even remotely to the nationwide level—failing in exchanges of sizes as low as 200 in the deterministic case (as shown by Dickerson et al. 2012b).

In this section, we augment the current UNOS solver to solve the failure-aware clearing problem on exchanges with edge failure probabilities. We first show that a powerful tool used in the current solver—the technique used to upper bound the objective value—is no longer useful. We show how to adapt the current solver's lower bounding technique to our model. We then significantly improve the core of the solver, which performs *column generation*, to only consider cycles and chains that are useful to the optimal failure-aware matching, and provide failure-aware heuristics for speeding up the column generation process.

5.1. Why We Cannot Use the Current UNOS Solver

In integer programming, a tree search that branches on each integral decision variable is used to search for an optimal solution. At each node, upper and lower bounds are computed to help prune subtrees and speed up the overall search. In practice, these bounding techniques are critical to proving optimality without exhaustively searching the space of all assignments.

5.1.1. Computing a Good Upper Bound Is Hard. The current kidney exchange solver uses the cycle cover problem with no length cap as a heuristic upper bound. This unrestricted clearing problem is solvable in polynomial time by encoding the pool into a weighted bipartite graph and computing the maximum weighted perfect matching (see reduction by Abraham et al. 2007). This is useful in practice because the unrestricted bound often matches the restricted (e.g., $|L| \leq 3$) optimal objective value. Unfortunately, for the failure-aware version of this problem, Proposition 2 shows that computing this bound is NP-hard.

Proposition 2. *The unrestricted failure-aware maximum cycle cover problem is NP-hard.*

Proof Sketch. We build on the proof of theorem 1 from Abraham et al. (2007), which shows that deciding if G admits a perfect cycle cover containing cycles of length at most 3 is NP-complete. They reduce from 3D-matching. All of the cycles in the constructed widgets in their proof are of length at least 3. Because of edge failures, a perfect cover that uses only 3-cycles has higher utility than any other cover, since each edge in a 3-cycle is worth more than a vertex in a k-cycle for $k > 3$ because of the all-or-nothing execution of cycles in a world with edge failure. The reduction of Abraham et al. (2007) has the property that there is a perfect cover with only 3-cycles if and only if there is a 3D-matching. Determining this is NP-complete, and thus the search problem is NP-hard. □

Driven by this hardness result, our new solver can use one of two looser upper bounds, solving the unrestricted clearing problem on a graph $G' = (V, E')$ with different edge weights or failure probabilities. In the (standard) setting where chains are executed partially until their first edge failure, the solver sets $q'_e = 1$, for each $e \in E$—that is, it solves the deterministic unrestricted clearing problem. Our experimental results in this paper assume partial execution of chains. For policies that allow chains to execute if and only if all edges in the chain exist, we set $w'_e = w_e q_e$, then $q'_e = 1$, for each $e \in E$, and solve the unrestricted clearing problem on that graph.

5.1.2. Computing a Good Lower Bound Is Not Hard.

The current UNOS solver uses the 2-cycle maximum matching problem (which is equivalent to the deterministic clearing problem for $L = 2$) as a primal heuristic, or lower bound. The new solver uses the *failure-aware* version of the 2-cycle maximum matching problem as a primal heuristic during the branch-and-price search. Solving this problem is still in polynomial time, as stated in Proposition 3.

Proposition 3. *The failure-aware clearing problem with cycle cap $L = 2$ is solvable in polynomial time.*

Proof. Given a directed compatibility graph $G = (V, E)$, construct an undirected graph $G' = (V, E')$ such that an edge exists between two vertices in G' if and only if there exists a 2-cycle between those vertices in G. Then, set the weight of every edge $e' = (v_i, v_j)$ in G' to

$$w_{e'} = q_{(v_i,v_j)} \cdot q_{(v_j,v_i)}(w_{(v_i,v_j)} + w_{(v_j,v_i)}).$$

Now find the maximum weighted matching on G', which can be done in polynomial time by Edmond's maximum-matching algorithm (1965).

5.1.3. Incremental Solving of Very Large IPs.

The number of decision variables in the integer program formulation of the clearing problem grows linearly with the number of cycles and chains in the pool. Unfortunately, the number of cycles grows polynomially in the cap L, and the number of chains grows exponentially! In fact, on pools generated using the state-of-the-art kidney exchange generator from Saidman et al. (2006), pools of size 5,000 containing *no chains* already contained nearly half a billion cycles. Including chains makes the full integer program impossible to store in memory.

Toward this end, the current UNOS solver uses an incremental formulation called column generation to bring only some variables into the search tree at each node. The basic idea behind column generation is to start with a reduced model of the problem, and then incrementally bring in variables (and their constraints) until the solution value of this reduced model is provably the solution value of the full (implicitly represented) model. This is done by solving the *pricing problem*, which associates with each variable a real-valued price such that, if any constraint in the full model for a variable c is violated, then the price of that variable is positive. In our case, the *price* of a cycle or chain c is just the difference between the expected utility $u(c)$ and the dual value sum of the vertices in that cycle or chain.

When no positive price cycles exist, we have proved optimality at this node in the search tree. Proving this is hard, since the solver might have to consider each cycle and chain. We now present a method for "cutting off" a chain after we know its expected utility is too low to improve the reduced problem's objective value.

5.2. Iterative Generation of Only Potentially "Useful" Chains

Given a k-chain $c = (v_0, v_1, \ldots, v_k)$, with v_0 an altruist, we show a technique for curtailing the generation of additions to c (while maintaining solution optimality). Consider the $(k+1)$-chain $c' = c + \{v_{k+1}\}$. Then the *additional* utility of this chain over c is just

$$\begin{aligned}
u(c') - u(c) &= \left(\sum_{i=1}^{k}(1-q_i)i\prod_{j=0}^{i-1}q_j + (k+1)\prod_{i=0}^{k}q_i\right) \\
&\quad - \left(\sum_{i=1}^{k-1}(1-q_i)i\prod_{j=0}^{i-1}q_j + k\prod_{i=0}^{k-1}q_i\right) \\
&= (1-q_k)k\prod_{i=0}^{k-1}q_i - k\prod_{i=0}^{k-1}q_i + (k+1)\prod_{i=0}^{k}q_i \\
&= (k+1)\prod_{i=0}^{k}q_i - q_k k\prod_{i=0}^{k-1}q_i \\
&= (k+1)\prod_{i=0}^{k}q_i - k\prod_{i=0}^{k}q_i = \prod_{i=0}^{k}q_i.
\end{aligned}$$

That is, the additional utility is just the probability of c' executing perfectly from start to finish (times the weight of the new edge, if $w_k \neq 1$).

Now, assume we are given some maximum success (minimum failure) probability $q_{\max}$ of the edges left in the remaining total pool of patients V' (so for $G = (V, E)$, the remaining pool is $V' = V \backslash c$). Then, an upper bound on the additional utility of extending c to an infinitely long chain c^∞ is just the geometric series:

$$u(c^\infty) - u(c) < \sum_{j=k}^{\infty} \prod_{i=0}^{k-1} q_i \prod_{i=k}^{j} q_{\max} = \prod_{i=0}^{k-1} q_i \left(\sum_{j=k}^{\infty} \prod_{i=k}^{j} q_{\max} \right).$$

Since $q_{\max} < 1$, this converges to

$$u(c^\infty) - u(c) =_{k\to\infty} \frac{q_{\max}}{1 - q_{\max}} \prod_{i=0}^{k-1} q_i. \tag{3}$$

An *upper bound* on the expected utility of a (possibly infinite) chain c', extended from some base k-chain $c = (v_0, v_1, \ldots, v_k)$, is given in Equation (3) above. We are interested in using this computed value to stop extending c.

Let the dual value of a vertex v be d_v. Furthermore, let $d_{\min}$ be the minimum dual value of any vertex in $V' = V - c$. Then a *lower bound* on the "cost" of using this extended chain c' is given by $d_{\min} + \sum_{i=0}^{k} d_i$.

By taking the optimistic upper bound on the utility of an infinite extension c' and the lower bound on the "cost" of using c', a criterion for c' not being useful is

$$\left(\frac{q_{\max}}{1 - q_{\max}} \prod_{i=0}^{k-1} q_i \right) + u(c) + l - \left(d_{\min} + \sum_{i=0}^{k} d_i \right) \leq 0. \tag{4}$$

Here, l is the utility derived from the final donor in a chain donating his or her kidney to the deceased donor waitlist. This is set by each individual kidney exchange.

Note that the sum of any finite subsequence of the infinite geometric series is less than the sum of the infinite series. Then, the first segment of Equation (4) can be only *lower* for any finite extension of c. Thus, if the inequality holds for the infinite extension, it must also hold for the finite extension.

Proposition 4. *Given a k-chain c, if the infinite extension c^∞ is not promising (i.e., Equation (4) holds), then no finite extension is promising, either.*

We use Proposition 4 to stop generating extensions of chains during our solver's iterative chain (column) generation routine. We incrementally maintain the expected utility of the chain $u(c)$ and the sum of the dual values of vertices in that chain, and compute the infinite series' convergence of the infinite chain whenever an extension is considered. If Equation (4) holds, from Proposition 4, we know no finite (or infinite) extension of c can have positive price, and the solver cuts off generating additions to c.

5.3. Heuristics for Generating Positive Price Chains and Cycles

During the column generation process, the optimizer iteratively brings positive price cycles and chains into a reduced linear program (LP). Once no cycles or chains outside the reduced LP have positive price, where the price of a cycle/chain c is defined to be $u(c) - \sum_{v \in c} d_v$, we can determine optimality from the reduced LP for the full LP.

In practice, the order in which positive price cycles and chains are brought into the reduced problem drives solution time. One approach is to try to generate those cycles and chains with lowest price. In our solver, we heuristically order the edges from which we start cycle or chain generation toward this end.

5.3.1. Ordering the Cycle Generation. For cycles, where v is a patient–donor vertex and v' is the vertex in v's outgoing neighbors with maximum failure-aware edge weight, we sort in descending order of ν:

$$\nu_v = \bar{q}_v^{in} q_{(v,v')} w_{(v,v')} - d_v.$$

Here, $\bar{q}_v^{in}$ is the average success probability of all incoming edges to v. Note that, for each vertex v, the $\bar{q}_v^{in} q_{(v,v')} w_{(v,v')}$ term can be computed exactly once (at cost $O(|V|^2)$), since these values do not change. Then, at each iteration of column generation, we perform an $O(|V| \log |V|)$ sort on the difference between this term and the current dual values.

Proposition 5. *For any nonaltruist v and next step v', such that $(q_{(v,v')} w_{(v,v')} - d_v) \leq 0$, we need not initiate cycle generation from v (which still guarantees all cycles are generated).*

Proof. A cycle c involves at least two vertices, including v. If v has a nonpositive dual-discounted weight, then at least one other vertex v' in the cycle must have positive dual-discounted weight. If not, the cycle will have nonpositive price and will not be considered in the column generation. Starting a search from v' will generate c.

5.3.2. Ordering the Chain Generation. For chains, where a is an altruist and v is the vertex corresponding to the initial edge from that altruist, we sort in descending order of ν:

$$\nu_{a,v} = q_{(a,v)} w_{(a,v)} - d_a.$$

The intuition here is that chains with a high utility outgoing edge (at low cost, from d_a) are more likely to be included in the final solution than those with low initial utilities. Note that we must consider all first hops out of all altruists, including those such that $\nu_{a,v} \leq 0$. Because of this, each iteration of column generation requires an $O(|A||V| \log(|A||V|))$ sort. With $|A|$ small, as in the UNOS exchange, this is an allowable cost.

6. Scalability Experiments

In this section, we test the ability of our new solver on kidney exchange compatibility graphs that are larger than current fielded kidney exchange pools, with an eye toward the future where kidney exchanges will be larger. We use data generated by the current state of the art kidney exchange instance generator by Saidman et al. (2006), augmented to include altruistic donors. These graphs are significantly denser than current kidney exchange pools. For a discussion on this, see Ashlagi et al. (2012) and Dickerson et al. (2012b). We test in the static (that is, myopic batch matching) setting here; in the next section, we expand to dynamic matching. For the experiments in this section, we assume a constant failure probability of 0.7 for each donor–patient edge.

We compare our novel solver against IBM ILOG CPLEX 12.2 (2010), a recent version of a state-of-the-art integer linear programming solver. Since CPLEX does not use branch-and-price, it must solve the full integer program (with one decision variable per possible cycle and chain).

Table 4 shows runtime and completion results for both solvers on graphs of varying size. Each graph has $|V|$ patient–donor pairs and $0.1|V|$ altruistic donors. For example, a row labeled $|V| = 50$ corresponds to a graph with 50 patient–donor pairs and 5 altruists. We generated 128 such graphs for each value of $|V|$. Each solver was allocated 8 GB of RAM and 1 hour of solution time on Blacklight, a large cc-NUMA shared-memory supercomputer at the Pittsburgh Supercomputing Center. (Blacklight was used solely to parallelize multiple runs for experimental results; our solver does not require any specialized hardware. In fact, the current version of our solver that runs the weekly matches at UNOS runs on commodity hardware.) CPLEX was unable to solve instances of size 100 (except once) in under an hour, while our solver was able to solve (at least some) instances of size 900.

To test how much speed was added by each of the improvements in this paper to the current UNOS solver, we deactivated the cycle and chain generation ordering heuristics (Section 5.3), as well as the solver's ability to cut off chain generation after the initial portion of a chain has been proven not to be in an optimal match (Section 5.2). Interestingly, removing the cycle and chain ordering heuristics did not noticeably affect the runtime or number of instances solved by our solver. Their low impact on performance is caused by the weak upper bounding performed during the IP solve; since the bounding is weak, often optimality must be proved by considering all (failure-discounted, possibly "good") chains and cycles, as opposed to being proved via bounding in the search tree. We believe these ordering heuristics, or ones like them, will hold greater merit when better bounding techniques are developed in the future. However, turning off the solver's ability to reason about the maximum additional expected utility of a chain did significantly affect overall runtime and number of instances solved; in fact, without this technique, only a single instance with 500 patient–donor pairs finished within the one hour time limit.

Table 4 also lists runtime results for those instances that did complete. When a solver was able to solve an instance within an hour, the solution time was typically quite low. This is a function of the upper and lower bounds becoming tight early on in the search tree. Overall, our method of incrementally generating cycles and chains results in dramatically increased completion percentages and lower runtimes than CPLEX.

7. A Model for Experimental Dynamic Kidney Exchange

In this section, we explore failure-aware matching in the context of dynamic kidney exchange. Kidney exchange is a naturally dynamic event, with patients, paired donors, and altruists arriving and departing

Table 4. Scaling Results for Our Method versus CPLEX, Timeout of 1 Hour, Reported Times in Seconds

	CPLEX		Ours		Ours without chain curtailing	
$\|V\|$	Solved	Time (solved)	Solved	Time (solved)	Solved	Time (solved)
10	127/128	0.044	128/128	0.027	128/128	0.052
25	125/128	0.045	128/128	0.023	128/128	0.049
50	105/128	0.123	128/128	0.046	125/128	0.057
75	91/128	0.180	126/128	0.072	123/128	0.066
100	1/128	1.406	121/128	0.075	121/128	0.071
150	0/128	—	114/128	0.078	95/128	0.098
200	0/128	—	113/128	0.135	76/128	0.096
250	0/128	—	94/128	0.090	48/128	0.133
500	0/128	—	107/128	0.264	1/128	0.632
700	0/128	—	115/128	1.071	0/128	—
900	0/128	—	38/128	2.789	0/128	—
1,000	0/128	—	0/128	—	0/128	—

Table 5. Reasons for the Arrival and Departure of Vertices and Edges

Vertex −	Edge −	Vertex/Edge +
Transplant, this exchange	Matched, positive crossmatch	Normal entrance
Transplant, deceased donor waitlist	Matched, candidate refuses donor	
Transplant, other exchange ("sniped")	Matched, donor refuses candidate	
Death or illness	*Pregnancy, sickness changes HLA*[a]	
Altruist runs out of patience		
Bridge donor reneges		

[a]We do not consider edge removal due to pregnancy/sickness because there are a variety of ways in which pregnancy and sickness can affect the immune system.

the pool over time. Section 4 enumerated some of the reasons we have seen in our experiences with the UNOS nationwide exchange. Formally, a dynamic kidney exchange can be explained by the evolution of its graph—that is, the addition and removal of its vertices and edges.

Table 5 formalizes the evolution of a compatibility graph over time. The only vertex and edge additions to the graph come in the form of new patients and donors arriving over time. Edges are removed because of, e.g., crossmatch failures or donor refusals. Vertices are removed if the patient or her respective donor must leave the pool, for reasons ranging from a successful transplantation to patient expiration.

Figure 6 provides a snapshot of a compatibility graph over three points in time. The pool at time t consists of unmatched patients and donors from time $t-1$, any new pairs and altruists entering the pool, and any vertices who were waiting for a successful match, but whose match failed (because of, e.g., a positive crossmatch). Note that these patients are still formally in the pool, just marked temporarily "inactive" until the status of their pending transplant is known. At each time period t, vertices leave the pool permanently through any of the reasons in the first column of Table 5, or are temporarily marked "inactive" through a pending match.

7.1. Failure-Aware Matching in Dynamic Kidney Exchange

We now present experimental results on dynamic kidney exchanges, taking transplant success probabilities into account. We built a simulator that mimics the evolutionary diagram of Figure 6, and used parameters learned from our work with UNOS. We vary the number of patient–donor pairs and altruists entering the pool over time, and match on a weekly basis for 24 weeks. We use the bimodal distribution of failure probabilities described in Section 4, as it more accurately represents current kidney exchanges. The deceased-donor waitlist donation at the end of a chain is counted in the expected number of transplants.

Figure 6. (Color online) The Evolution Dynamics of a Kidney Exchange

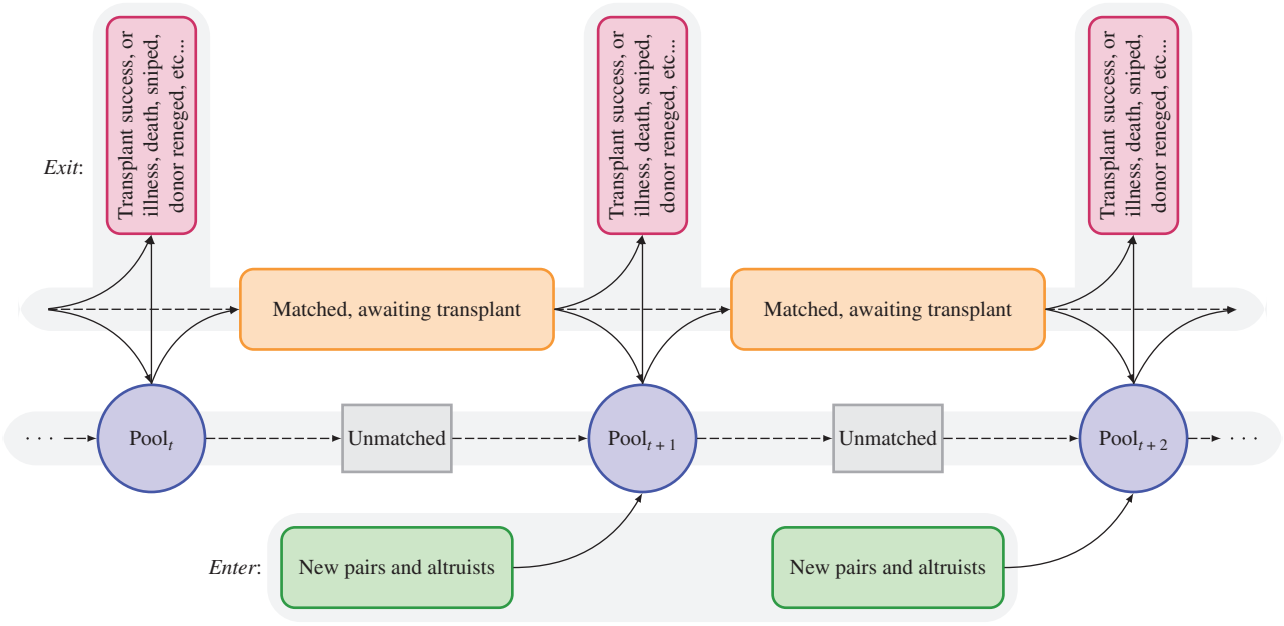
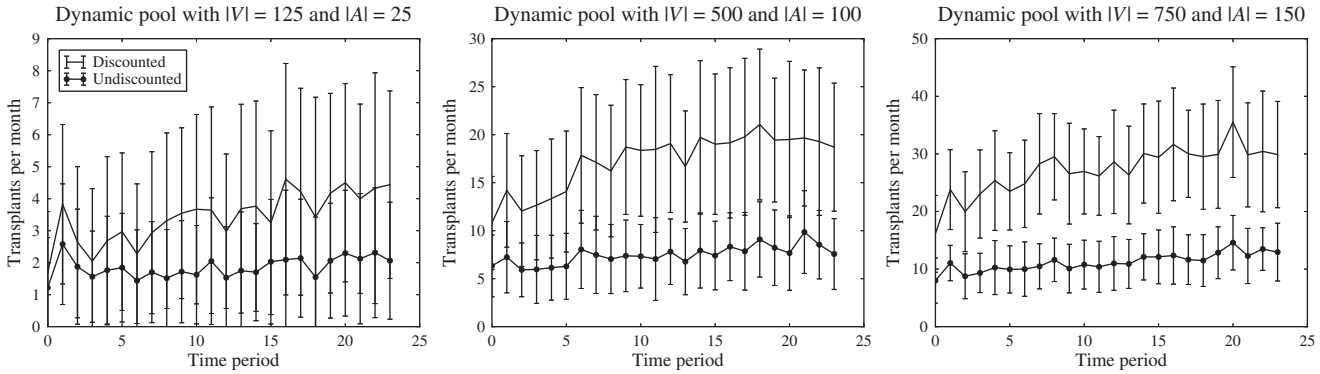


Figure 7. Expected Number of Transplants per Week for Graphs of Different Sizes



Note. From left to right, five pairs and one altruist, 20 pairs and four altruists, and 25 pairs and five altruists (on expectation) appear every week.

In our experience with UNOS, typically the time between a match offer and successful transplant is about 8 weeks. Thus, whenever a match is offered in our simulator, involved patients and donors become inactive in the pool but can still be removed from the match for a variety of reasons ("sniping" by another exchange, patient illness, etc.). Of the 610 patients who had ever been listed in the UNOS exchange program when these experiments were run (over a period of 106.7 weeks), 192 left for reasons other than receiving a kidney through UNOS. Thus, for each time period, a vertex has a probability of $1 - e^{(\ln 418/610)/106.7} \approx 0.003536$ chance of leaving (for a non-UNOS transplant reason). As in real kidney exchange, if a cycle fails, or part of a chain fails, then the affected patients and donors are returned to the pool. However, if the reason for failure was that patient–donor pair's exit from the exchange, that vertex is removed permanently, along with all incident edges. Results from crossmatches that were done as part of a failed cycle or chain are maintained in the pool; if a crossmatch was negative, then future crossmatches performed on that edge will also be negative. We assume that all crossmatches are done simultaneously for cycles and incrementally from the initiating nondirected donor until the first failure for chains.

Figure 7 shows the number of expected transplants per week on graphs of three different sizes, each generated from the Saidman et al. (2006) distribution of compatibility graphs. (In the following section, we generate graphs from the UNOS distribution.) In expectation, 5, 20, or 25 pairs and 1, 4, or 5 altruists appear weekly in each of the three graphs. Failure-aware matching typically results in roughly twice as many expected transplants than maximum cardinality matching. The slight increase in weekly expected matches for both matching techniques is due to the buildup of unmatched patient–donor pairs and altruists in the pool over time; larger pools typically admit larger matchings.

Figure 8. Expected Aggregate Transplants Over 24 Weeks, for Increasing $|V|$ (and $|A| = 0.1|V|$)

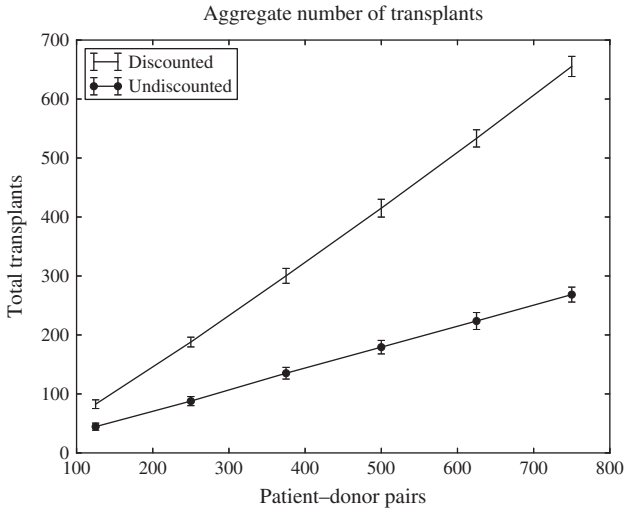


Figure 8 gives aggregate results for total number of expected transplants over 24 weeks, for graphs of varying size, for both failure-aware and maximum cardinality matching. Graphs have 10% as many altruists on top of the patient–donor pool. The gap between failure-aware and non-failure-aware matching widens as the activity level of the dynamic kidney exchange increases. For our largest graphs, failure-aware matching improved expected transplants by a factor of three over maximum cardinality matching. In the following section, we will explore how these global efficiency gains change as we prioritize highly sensitized patients and on graph distributions that more closely mimic presently fielded exchanges.

8. Balancing Efficiency and Fairness in Failure-Aware Kidney Exchange

So far, we have motivated a move to failure-aware kidney exchange optimization from a global efficiency

perspective. One might ask how this affects fairness. For example, a proposed transplant to a highly sensitized patient might intuitively fail with higher probability than one to a patient of low sensitization due to coupled health issues (e.g., chronic illness) in the former, and thus the failure-aware approach could disfavor highly sensitized patients. While data from the UNOS kidney exchange (Kidney Paired Donation Work Group 2013) do not show a correlation between postmatch failure and CPRA, data from other exchanges do show such a correlation (e.g., Ashlagi et al. 2011, Glorie 2012). Regardless, prioritizing of highly sensitized patients is currently done explicitly or implicitly in fielded kidney exchanges, so we address that here.

In general, striking a balance between fairness and efficiency in kidney exchange is an increasingly important line of work combining medical policy, economics, and optimization. Roth et al. (2005b) define a fair mechanism to be one that equalizes, to the greatest extent possible, patients' chances of getting a match. While this is almost certainly too strict a fairness criterion to be fielded in practice, the notion of prioritizing some patients—possibly at the cost of overall efficiency in the exchange—is common (and is performed in the current UNOS exchange as well). Recent and parallel work by Bertsimas et al. (2011, 2012) and by Caragiannis et al. (2012) studies the *price of fairness*, a measure of the trade-off between fairness and efficiency, in general resource allocation problems. Hooker and Williams (2012) provide general Rawlsian equity optimization models that maximize the minimum utility of any one agent or set of agents. Bertsimas et al. (2013) design a realistic method for maximizing, given a set of user-defined fairness constraints, some notion of efficiency in the *deceased-donor* kidney transplantation problem, where patients on a waiting list are allocated cadaveric kidneys. In general, accurate quantification of the theoretical and empirical advantages and disadvantages of various fairness definitions would be of great value to policymakers in the kidney exchange community.

In this work, we adapt a fairness criterion from Dickerson et al. (2014b), who investigated the price of fairness in kidney exchange. They proved analytically that the price of fairness in a static, deterministic, simplified model of kidney exchange is low. In the rest of this section, we show (experimentally) that the price of fairness in both static and dynamic failure-aware models is also typically low. More importantly, we show that failure-aware matching under well-chosen fairness criteria results in more expected transplants to *both* the global pool *and* highly sensitized patients than maximum cardinality matching. We conclude that there is an enormous "price of using the wrong model" that is potentially more harmful to *all* patients.

8.1. Weighted Fairness as a Prioritization Scheme for Sensitized Patients

One simple method to emphasize a certain class of patient–donor pairs—for us, those in the set of highly sensitized vertices V_H—is to increase the weight of edges with a sink in V_H. This definition generalizes the policy UNOS currently applies to highly sensitized patients in the fielded kidney exchange, where incoming edges to patients above a certain CPRA threshold are given a positive constant additive weight increase. We adopt a parameterized form of this rule here.

To implement this rule, Dickerson et al. (2014b) build on the standard kidney exchange integer programming formulation and rewrite the objective as follows:

$$\max \sum_c v_\Delta(c) x_c.$$

Here, $v_\Delta(c)$ is the value of a cycle or chain c (either the weight in the deterministic model or the expected utility in our failure-aware model) such that the weight of each edge $e \in c$ is adjusted by some reweighting function $\Delta: E \to \mathbb{R}$.

A simple example reweighting function is multiplicative:

$$\Delta^\beta(e) = \begin{cases} (1+\beta)w_e & \text{if } e \text{ ends in } V_H, \\ w_e & \text{otherwise.} \end{cases}$$

Intuitively, for some $\beta > 0$, this function scales the weight of edges ending in highly sensitized vertices by $(1+\beta)$. For example, if $\beta = 0.5$, then the optimization algorithm will value edges that result in a highly sensitized patient receiving a transplant at 50% above their initial weight (which may then be discounted by other factors like failure probability and chain position, as in our paper's current model).

For any $M \in \mathcal{M}$, let M' be the matching such that every cycle $c \in M$ has augmented weight $v_\Delta(c)$—that is, M' is the same set of cycles included in the initial matching M, only with utilities associated with those cycles in accordance with reweighting function Δ. Then define the *weighted* fairness rule u_Δ in terms of the utilitarian rule u applied to the augmented matching M', such that $u_\Delta(M) = u(M')$. Thus, the clearing problem is rewritten as finding $M^* = \arg\max_{M \in \mathcal{M}} u_\Delta(M)$.

In the rest of this section, we explore the effect that this weighted fairness rule has on the expected number of transplants performed in the pool as a whole and by highly sensitized patients in V_H, under a variety of modeling assumptions.

8.2. Experiments in the Static Setting

We begin by studying the weighted fairness rule in the context of static kidney exchange. We do this both on the 161 individual UNOS match runs to date and on generated graphs that mimic the UNOS graphs.

The generator runs by loading all pairs and altruistic donors that have ever been present in the UNOS pool into a set of vertices V, then drawing with replacement vertices from that pool and running the UNOS edge existence algorithm on the sampled vertices to create a compatibility graph. We test these real or sampled graphs under three probability distributions: constant and bimodal as above, as well as a differently distributed bimodal family that draws failure probabilities in accordance with those rates published by Ashlagi et al. (2011). Critically, this last distribution correlates edge failure rate with patient CPRA; incoming edges to highly sensitized patients are much more likely to fail than incoming edges to the rest of the pool. Specifically, they state that patients with a CPRA above 75 have a crossmatch failure probability of 0.5, while those with lower CPRA values (reported in ranges [0–24], [25–49], and [50–74]) have much lower probabilities of crossmatch failure (0.05, 0.2, and 0.35, respectively). They also experiment with an additional additive exogenous failure rate varied between 0 and 0.16; we use 0.08 in our experiments.

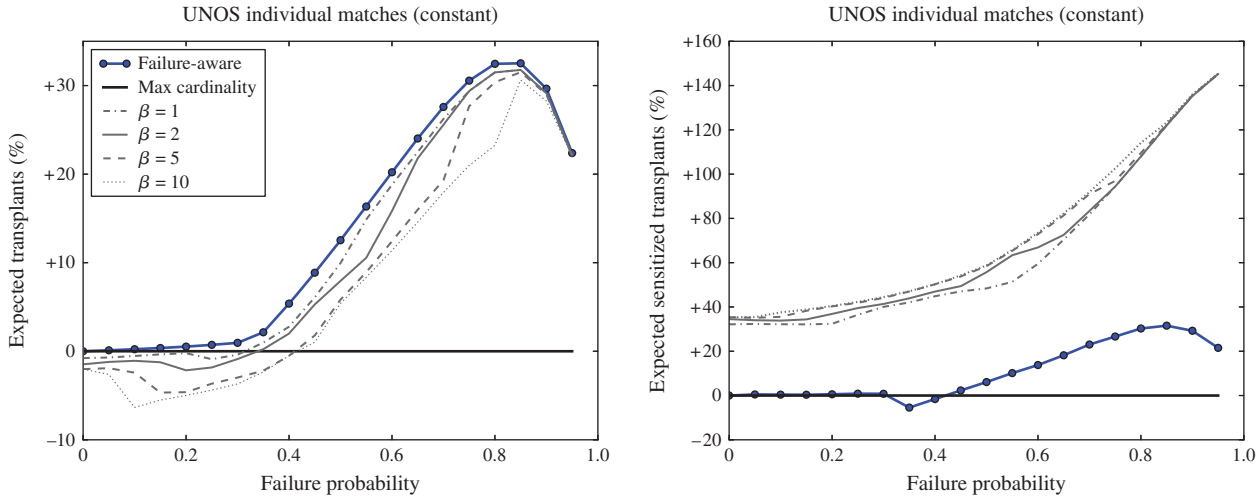
8.2.1. Constant Failure Rate. We begin by assuming that every edge fails with the same constant probability, as in previous sections. This assumption, while not likely to hold in practice, is easily parameterized and allows us to explore the differences in models as matchings become less reliable. Different exchanges have different failure rates, and this exploratory analysis might serve as a useful tool to quantify the marginal gains of decreasing edge failure rates.

Figure 9 compares the weighted fairness rule u_Δ applied to the failure-aware model against the utilitarian rule applied to the deterministic model, which computes a maximum cardinality disjoint cycle cover without regard for edge failure. The left-hand panel of Figure 9 shows that the efficient failure-aware matching always results in at least as many (typically more) expected transplants as the efficient deterministic matching. However, interestingly, even matchings under the fair rule u_Δ in the failure-aware model often result in significant overall gains when compared to the utilitarian deterministic matching. The right-hand panel of Figure 9 shows that even the *fully efficient* matching rarely results in a loss of highly sensitized transplants, and that even slightly prioritizing sensitized patients results in large gains (at low cost to global efficiency).

For example, for $\beta = 1.0$ (that is, when highly sensitized patients are valued twice as much as lowly sensitized patients), we see a drop of only a couple of percentage points of expected transplants when there is *no* probability of edge failure. This is countered by a very large (over 30%) gain in the expected number of highly sensitized transplants. In fact, when the probability of edge failure is at least 45%, valuing highly sensitized transplants at 11× ($\beta = 10.0$) that of a lowly sensitized patient results in more expected total transplants than deterministic matching that does not consider fairness.

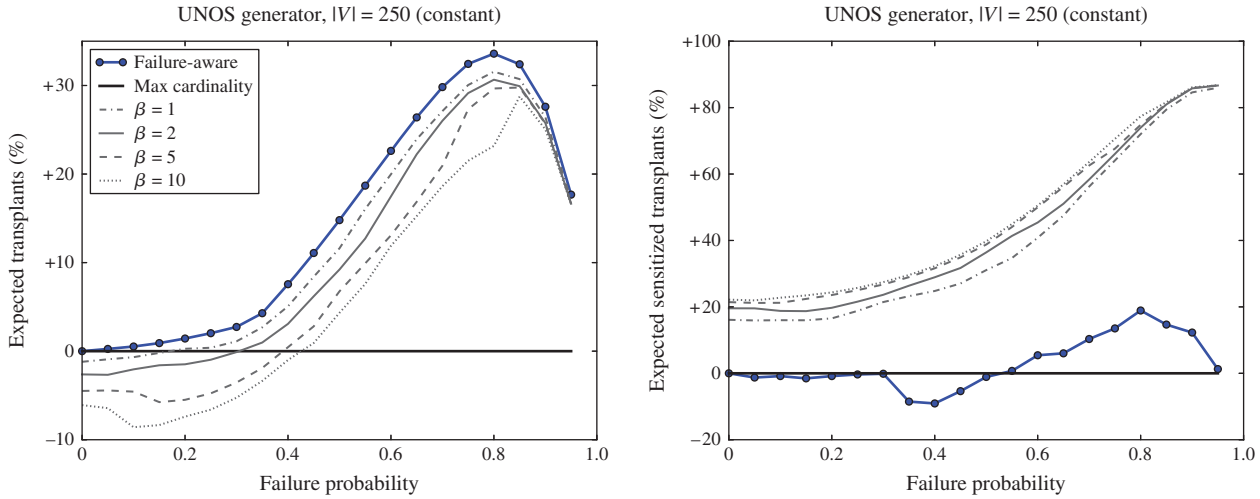
Also, we see that efficient failure-aware matching almost always results in more expected sensitized transplants than deterministic matching, with the exception of a small relative drop at failure rates around 35%–45%. This can be explained by comparing, given a failure probability p, the relative expected utilities of a 2-cycle c_2 ($u(c_2) = 2(1-p)^2$) and 3-cycle c_3 ($u(c_3) = 3(1-p)^3$). When $p < \frac{1}{3}$, $u(c_2) < u(c_3)$, so the optimizer favors 3-cycles over 2-cyles. When $p > \frac{1}{3}$, $u(c_2) > u(c_3)$, so the optimizer favors 2-cycles. Highly

Figure 9. (Color online) Percentage Change in Expected Number of Transplants (Left) and Sensitized Transplants (Right) for *Actual* UNOS Match Runs Using Failure-Aware Matching—Possibly with Fairness Constraints—Instead of Maximum Cardinality Matching



Note. The x axis varies constant edge failure probability from zero to near one.

Figure 10. (Color online) Percentage Change in Expected Number of Transplants (Left) and Sensitized Transplants (Right) for *Generated* UNOS Match Runs Using Failure-Aware Matching—Possibly with Fairness Constraints—Instead of Maximum Cardinality Matching



Note. The x axis varies constant edge failure probability from zero to near one.

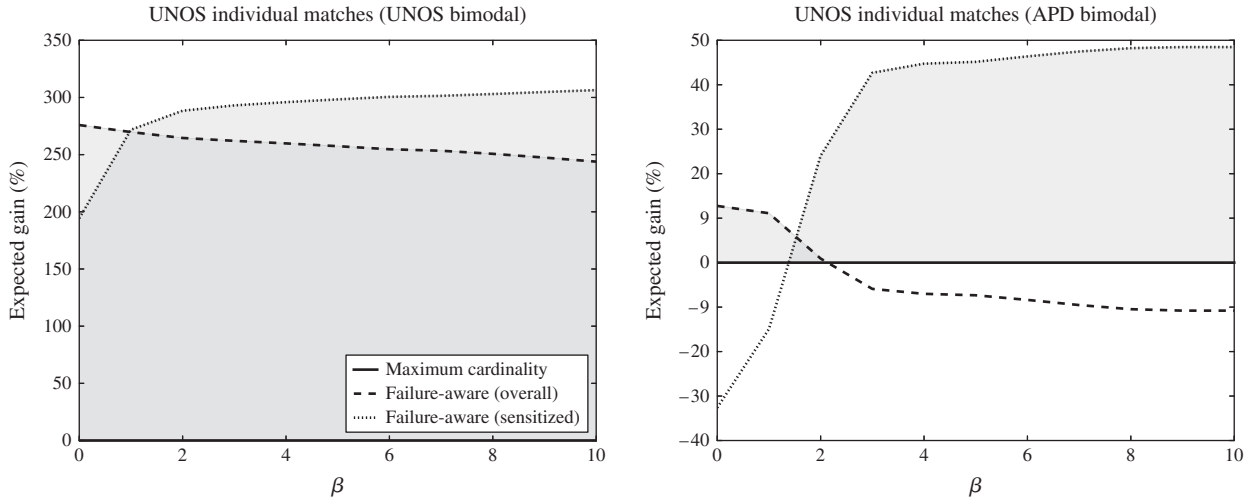
sensitized patients are often matched in 3-cycles; intuitively, if a highly sensitized pair's donor can donate to another pair, it is more likely that this pair will not be able to connect back to the highly sensitized pair directly (by virtue of that initial pair being highly sensitized and thus having low in-degree) via a 2-cycle but will rather connect back through a lowly sensitized pair via a 3-cycle). So, for $p < \frac{1}{3}$, failure-aware gains are only realized by rearranging the low-probability tails of chains into 2- and 3-cycles, while for $p > \frac{1}{3}$, failure-aware optimization may start to cannibalize 3-cycles (that likely contain highly sensitized pairs). Empirically, this is only an issue for $p \in (\frac{1}{3}, 0.45]$; once $p > 0.45$, the efficient objective's gains outweigh these losses. Furthermore, we see that a small prioritization (even $\beta = 1$) results in both global and sensitized gains even for $p \in (\frac{1}{3}, 0.45]$ (and for other values of p).

This general behavior is supported in Figure 10, which shows the same experiments on generated data that mimic the UNOS distribution, for pools of size 250—roughly the size of the current UNOS pool. We include these results because, in Section 8.3, we run dynamic experiments on data that mimic the UNOS pool (unlike the results in Section 7, which used the Saidman et al. 2006 generator). The similarity of Figures 9 and 10 serves as validation of the simulator.

It may be difficult to accurately estimate failure probabilities on edges in practice. Indeed, in extreme cases, it may even be deemed unethical to allow vastly different failure probabilities to be included in the optimization process, as the probabilities act as a prioritization tool. As these experiments show, one could simply set all of the probabilities in the optimization to be equal in order to not disfavor patients with high failure probabilities. Even with this extreme approach, the failure-aware framework strikes good endogenous trade-offs between short chains, long chains, short cycles, and long cycles—unlike the current deterministic approach.

8.2.2. Bimodal Failure Rate. We now consider the weighted fairness rule in the static setting with bimodal failure probabilities. We will refer to the prior bimodal failure distribution derived in Section 4, where edge failure rates are not correlated with patient CPRA, as the "UNOS Bimodal" distribution. We also perform experiments on a distribution derived from published failure rates from a different exchange, the Alliance for Paired Donation (APD), where edge failures *are* correlated with patient CPRA (Ashlagi et al. 2011). We refer to this distribution as "APD Bimodal." This difference in correlations could be due to highly sensitized patients being less likely to find a match outside of the exchange (e.g., on the deceased-donor wait list or another exchange) but more likely to have a match fail because of medical reasons such as crossmatch incompatibility—whereas an easy-to-match patient might quickly find a living donor elsewhere but be less likely to have a match fail for medical reasons. UNOS has a slower matching cadence than some other exchanges like the National Kidney Registry (NKR), which matches whenever the underlying compatibility graph changes, so easily matched patients may be "sniped" by such faster-moving exchanges. By lowering these nonmedical reasons for failure (e.g., by merging all exchanges into a single program to reduce interexchange competition), the overall failure rate for highly sensitized patients would probably become higher than that of other patients.

Figure 11. Change in the Expected Number of Transplants on Average for *Actual* UNOS Match Runs When Using Failure-Aware Matching Instead of Maximum Cardinality Matching, Assuming Bimodal Edge Failure Rates Derived from UNOS (Left) and APD (Right)



Note. The x axis varies the β fairness factor applied to the failure-aware matching algorithm.

Figure 11 shows expected gains in both the number of overall transplants (dashed line) and sensitized transplants (dotted line) relative to a baseline of deterministic matching (solid line). The expected number of failure-aware overall and highly sensitized transplants are compared against the expected number of deterministic overall and highly sensitized transplants, respectively, as the fairness factor β is increased from 0 (fully efficient matching) to 10 (highly biased matching).

Immediately visible is that, when failure rates are not correlated to CPRA, the gains seen by failure-aware matching are quite large across the board. This aligns with our Saidman-generated results from Section 7, as well. However, when failure rates are highly correlated with patient CPRA, the situation becomes more delicate. Failure-aware matching without fairness considerations does result in a large gain in overall expected transplants, but harms highly sensitized patients. We can identify a "sweet spot" that balances these conflicting objectives; empirically, this is approximately when $\beta \in [2, 4]$. When β is toward the lower end of this interval, the loss in marginalized transplants is zero while the gain in global expected transplants is positive (approximately 10%). When β is toward the higher end of this range, the global gain in transplants is zero while the gain in marginalized transplants is positive (approximately 25%). Within the interval, we realize gains in both objectives—a clear win.

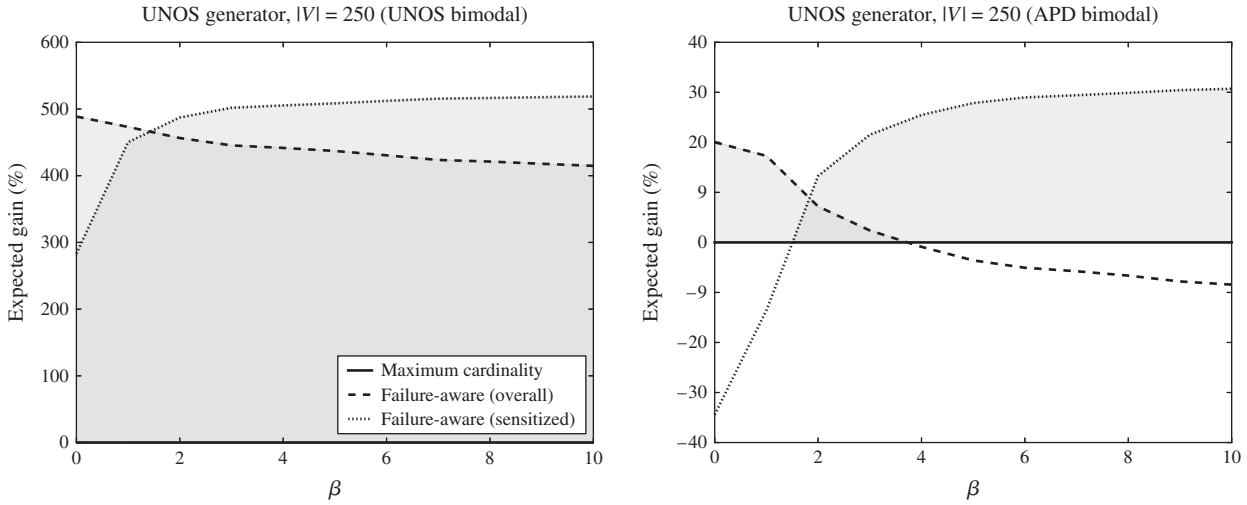
As in the constant failure probability case, Figure 12 shows similar results on generated UNOS compatibility graphs, under both failure rate distributions, for $|V| = 250$. This provides validation for our simulator. In the rest of the section, we further explore the correlated failure rate setting in the realistic dynamic kidney exchange simulator presented in Section 7 using these equally realistic compatibility graphs, and show that this same balance of fairness and efficiency can be struck so that *both* global efficiency and the expected number of transplants to highly sensitized patients increases.

8.3. Experiments in the Dynamic Setting

We now continue our exploration of the correlated failure probability case into a dynamic model. This is important because, although we showed that a balance can be struck between efficiency and fairness in the static case such that failure-aware matching results in gains in both objectives, it is possible that this balance comes at the cost of matching "easier" hard-to-match pairs in the now and leaving the "hardest" hard-to-match pairs for later. We show that this is not the case. Specifically, the same winning balance can be struck in the dynamic setting. (In the interest of space, we do not include experiments in the noncorrelated bimodal failure case, because even failure-aware matching without fairness considerations results in large increases in both global and marginalized transplants over time. In this sense, the experiments in this section on the correlated APD distribution give a conservative estimate of the gains seen by failure-aware matching in dynamic kidney exchange.)

We perform experiments in the same dynamic model as Section 7, only this time using the realistic UNOS graph generator validated above. We vary arrival rates over 24 time periods with $\{12, 16, \ldots, 32\}$ pairs or altruistic donors arriving per time period, as sampled from the real pairs and altruists. Tables 6 and 7 show the median overall absolute and percentage gains and

Figure 12. Change in the Expected Number of Transplants on Average for *Generated* UNOS Match Runs When Using Failure-Aware Matching Instead of Maximum Cardinality Matching, Assuming Bimodal Edge Failure Rates Derived from UNOS (Left) and APD (Right)



Note. The x axis varies the β fairness factor applied to the failure-aware matching algorithm.

losses in number of transplants and number of sensitized transplants, respectively, aggregated over all time periods by failure-aware matching for $\beta \in \{0, 1, \ldots, 5\}$ compared against deterministic matching.

Mirroring the static experiments above, we see that for low values of β, failure-aware matching results in global gains and marginalized losses. However, as above, for $\beta \approx 2$, a winning balance is struck, with nonnegative gains in expected overall transplants and significant gains in number of highly sensitized transplants. Perhaps most excitingly, for higher values of β, the number of highly sensitized transplants increases markedly (reaching +20%–+40% over deterministic matching for higher arrival rates), while the overall effect on global efficiency is negligible. In reality, kidney exchanges are often seen as a "last hope" for

Table 6. Gains in Expected Number of Transplants Overall, for Increasing Values of Fairness β and for Different Arrival Rates

	$\lvert V\rvert = 300$		$\lvert V\rvert = 400$		$\lvert V\rvert = 500$		$\lvert V\rvert = 600$		$\lvert V\rvert = 700$		$\lvert V\rvert = 800$	
	Gain	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)
Efficient	+0	(0.0)	+5	(5.9)	+1	(1.9)	+2	(2.5)	+9	(7.1)	+5	(3.6)
Fair, $\beta = 1$	+2	(4.2)	+5	(6.7)	+1	(1.0)	+8	(8.1)	+8	(6.2)	+11	(7.3)
Fair, $\beta = 2$	+0	(0.0)	+3	(4.1)	+0	(−1.3)	+3	(2.4)	+2	(1.8)	+5	(3.4)
Fair, $\beta = 3$	+2	(4.3)	−1	(−2.1)	−1	(−1.1)	−1	(−1.3)	+3	(2.8)	+2	(1.5)
Fair, $\beta = 4$	+2	(4.3)	+2	(2.5)	+2	(2.5)	−1	(−1.3)	+1	(0.9)	+3	(2.3)
Fair, $\beta = 5$	+0	(−0.1)	+1	(2.0)	+3	(4.0)	+0	(−0.5)	−1	(−0.8)	−2	(−1.7)

Table 7. Gains in Expected Number of Highly Sensitized Transplants, for Increasing Values of β and for Different Arrival Rates

	$\lvert V\rvert = 300$		$\lvert V\rvert = 400$		$\lvert V\rvert = 500$		$\lvert V\rvert = 600$		$\lvert V\rvert = 700$		$\lvert V\rvert = 800$	
	Gain	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)
Efficient	−4	(−40.0)	−2	(−21.4)	−3	(−15.4)	−4	(−21.4)	−5	(−23.4)	−6	(−19.1)
Fair, $\beta = 1$	−2	(−26.1)	+0	(0.0)	−1	(−10.0)	+0	(0.0)	+0	(−1.3)	−1	(−4.4)
Fair, $\beta = 2$	+1	(9.5)	+3	(18.8)	+0	(1.2)	+2	(9.9)	+2	(11.2)	+5	(15.5)
Fair, $\beta = 3$	+0	(5.6)	+1	(10.8)	+1	(11.7)	+7	(35.1)	+8	(33.2)	+6	(20.3)
Fair, $\beta = 4$	+0	(5.6)	+3	(29.0)	+2	(11.0)	+8	(46.2)	+6	(23.9)	+8	(29.3)
Fair, $\beta = 5$	+0	(0.0)	+2	(22.6)	+2	(12.1)	+8	(43.7)	+6	(24.0)	+8	(23.9)

highly sensitized patients; even with a higher likelihood of pretransplant match failure, we have shown that failure-aware matching can increase successful match rates for these highly prioritized patients at no cost to the global system efficiency.

9. Conclusions and Future Work

In this paper, we addressed the problem of edges in a matching (e.g., recommended transplants in a kidney exchange) failing after a matching algorithm has committed to them. This is a timely problem; in the UNOS nationwide kidney exchange, only 7% of algorithmically matched patients actually receive a transplanted kidney through the exchange, and similar rates apply to other kidney exchanges. We introduced a failure probability to each edge in a compatibility graph, and defined an expected utility of edges, cycles, chains, and matches. This model drives our main theoretical result, that (with high probability, in a random graph model) there exists a nonmaximum cardinality matching that provides linearly more utility than any maximum cardinality matching. We then ran simulations on real data from all UNOS match runs between 2010 and late 2014, and found that our failure-aware matching increases the number of expected transplants dramatically. Critically, we also found that this result is robust to uncertainty over the true underlying failure rates—an uncertainty that exists in reality.

Armed with this new model, we showed that the current state-of-the-art kidney exchange solver (used in the UNOS kidney exchange) cannot be used for this problem because now each edge has both a weight and a failure probability, and simply multiplying them to get a revised weight would make the algorithm incorrect. We designed a branch-and-price–based optimal clearing algorithm specifically for the probabilistic exchange clearing problem. It has many enhancements over the prior best kidney exchange clearing algorithm. For one, we designed a failure-aware column generator that incrementally brings only "possibly good" chains into consideration. We showed experimentally that this new solver scales well on large simulated data. We then developed a realistic model of dynamic kidney exchange based on our experiences with, and data from, UNOS, and showed that failure-aware matching in dynamic graphs increases expected transplants significantly. Finally, we explored the effect of failure-aware matching on marginalized patients. This led to the main practical result of this paper: that it is possible to strike a balance between fair and efficient failure-aware matching that results in more expected transplants *both* globally and to marginalized patients specifically, in both the static and dynamic cases, in a variety of graph distributions.

Acknowledgments

The authors acknowledge Intel Corporation and IBM for gifts. The authors thank the anonymous reviewers at EC-13 and participants of the IBM HCAGT workshop, as well as the anonymous reviewers of this journal paper. The authors also thank Ruthanne Leishman, Elizabeth Sleeman, Darren Stewart, and the rest of the UNOS KPD Pilot Program staff.

Endnotes

[1] This paper is a significant extension of Dickerson et al. (2013a), and includes new computational ideas, techniques, and results on real kidney exchange data from the UNOS nationwide kidney exchange from its inception in 2010 through November of 2014 and on simulated data at sizes greater than any current fielded kidney exchange. It also benefits from extensive discussion with surgeons and economists at the 2013 American Transplant Congress and 2014 World Transplant Congress (especially regarding work by Leishman et al. 2013 and Dickerson et al. 2013b, 2014a).

[2] Another challenge in kidney exchanges is that transplant centers hide some of their donor–patient pairs and altruistic donors from the exchange and instead try to match them locally. This is a major problem in practice. For example, of the pairs revealed to the UNOS exchange from its beginning in October 2010 to May 2012, *none* could have been locally matched in their transplant centers (Stewart et al. 2013). In other words, the centers did not reveal any of their pairs that could be locally matched to the exchange. There is no perfect mechanism design solution to that problem (see, e.g., Ashlagi and Roth 2014, Ashlagi et al. 2015, Sönmez and Ünver 2013). Still, the only way to motivate the centers to fully reveal their pairs and altruists is by mandate, and it is not clear that is politically viable. This paper does not address this problem, except to the extent that better matching generally speaking gives more motivation for the centers to participate because success chances for their patients become better and wait times shorter.

[3] The aggregate match data from which we infer crossmatch failure probabilities is available in a report from the Kidney Paired Donation Work Group (2012) and summarized by Leishman et al. (2013). Updated aggregate data are now available in a report from the Kidney Paired Donation Work Group (2013); these most recent data were not incorporated into our experiments, but they are very similar to those that were.

References

Abraham D, Blum A, Sandholm T (2007) Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. *Proc. 8th ACM Conf. Electronic Commerce* (ACM, New York), 295–304.

Akbarpour M, Li S, Gharan SO (2014) Dynamic matching market design. *Proc. Fifteenth ACM Conf. Econom. Comput.* (ACM, New York), 355–355.

Anderson R (2014) Stochastic models and data driven simulations for healthcare operations. PhD thesis, Massachusetts Institute of Technology, Cambridge.

Anderson R, Ashlagi I, Gamarnik D, Kanoria Y (2015a) A dynamic model of barter exchange. *Annual ACM-SIAM Sympos. Discrete Algorithms* (ACM, New York), 1925–1933.

Anderson R, Ashlagi I, Gamarnik D, Roth AE (2015b) Finding long chains in kidney exchange using the traveling salesman problem. *Proc. Natl. Acad. Sci. USA* 112(3):663–668.

Anshelevich E, Chhabra M, Das S, Gerrior M (2013) On the social welfare of mechanisms for repeated batch matching. *Proc. Twenty-Seventh AAAI Conf. Artificial Intelligence* (AAAI, Menlo Park, CA), 60–66.

Ashlagi I, Roth AE (2014) Free riding and participation in large scale, multi-hospital kidney exchange. *Theoret. Econom.* 9(3):817–863.

Ashlagi I, Jaillet P, Manshadi VH (2013) Kidney exchange in dynamic sparse heterogenous pools. *Proc. Fourteenth ACM Conf. Electronic Commerce* (ACM, New York), 25–26.

Ashlagi I, Fischer F, Kash IA, Procaccia AD (2015) Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. *Games Econom. Behav.* 91(May):284–296.

Ashlagi I, Gamarnik D, Rees M, Roth AE (2012) The need for (long) chains in kidney exchange. NBER Working Paper 18202, National Bureau of Economic Research, Cambridge, MA.

Ashlagi I, Gilchrist DS, Roth AE, Rees M (2011) Nonsimultaneous chains and dominos in kidney-paired donation—Revisited. *Amer. J. Transplantation* 11(5):984–994.

Assadi S, Khanna S, Li Y (2016) The stochastic matching problem with (very) few queries. *Proc. 2016 ACM Conf. Econom. Comput.* (ACM, New York), 43–60.

Awasthi P, Sandholm T (2009) Online stochastic optimization in the large: Application to kidney exchange. *Proc. 21st Internat. Joint Conf. Artificial Intelligence* (*IJCAI*), *Pasadena, CA*, 405–411.

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* 46(3):316–329.

Bertsimas D, Farias VF, Trichakis N (2011) The price of fairness. *Oper. Res.* 59(1):17–31.

Bertsimas D, Farias VF, Trichakis N (2012) On the efficiency-fairness trade-off. *Management Sci.* 58(12):2234–2250.

Bertsimas D, Farias VF, Trichakis N (2013) Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Oper. Res.* 61(1):73–87.

Biró P, Manlove DF, Rizzi R (2009) Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Math. Algorithms Appl.* 1(4):499–517.

Blum A, Gupta A, Procaccia AD, Sharma A (2013) Harnessing the power of two crossmatches. *Proc. Fourteenth ACM Conf. Electronic Commerce* (ACM, New York), 123–140.

Blum A, Dickerson JP, Haghtalab N, Procaccia AD, Sandholm T, Sharma A (2015) Ignorance is almost bliss: Near-optimal stochastic matching with few queries. *Proc. Sixteenth ACM Conf. Econom. Comput.* (ACM, New York), 325–342.

Bray M, Wang W, Song PX-K, Leichtman AB, Rees MA, Ashby VB, Eikstadt R, Goulding A, Kalbfleisch JD (2015) Planning for uncertainty and fallbacks can increase the number of transplants in a kidney-paired donation program. *Amer. J. Transplantation* 15(10):2636–2645.

Caragiannis I, Kaklamanis C, Kanellopoulos P, Kyropoulou M (2012) The efficiency of fair division. *Theory Comput. Systems* 50(4): 589–610.

Chen Y, Li Y, Kalbfleisch JD, Zhou Y, Leichtman A, Song PX-K (2012) Graph-based optimization algorithm and software on kidney exchanges. *IEEE Trans. Biomedical Engrg.* 59:1985–1991.

Dickerson JP, Sandholm T (2014) Multi-organ exchange: The whole is greater than the sum of its parts. *Proc. Twenty-Eighth AAAI Conf. Artificial Intelligence* (AAAI, Menlo Park, CA), 1412–1418.

Dickerson JP, Sandholm T (2015) FutureMatch: Combining human value judgments and machine learning to match in dynamic environments. *Proc. Twenty-Ninth AAAI Conf. Artificial Intelligence* (AAAI, Menlo Park, CA), 622–628.

Dickerson JP, Procaccia AD, Sandholm T (2012a) Dynamic matching via weighted myopia with application to kidney exchange. *Proc. Twenty-Sixth AAAI Conf. Artificial Intelligence* (AAAI, Menlo Park, CA), 1340–1346.

Dickerson JP, Procaccia AD, Sandholm T (2012b) Optimizing kidney exchange with transplant chains: Theory and reality. *Proc. 11th Internat. Conf. Autonomous Agents and Multi-Agent Systems* (AAMAS, Richland, SC), 711–718.

Dickerson JP, Procaccia AD, Sandholm T (2013a) Failure-aware kidney exchange. *Proc. Fourteenth ACM Conf. Electronic Commerce* (ACM, New York), 323–340.

Dickerson JP, Procaccia AD, Sandholm T (2013b) Results about, and algorithms for, robust probabilistic kidney exchange matching. *Amer. Transplant Congress* (*ATC*). Poster abstract.

Dickerson JP, Procaccia AD, Sandholm T (2014a) Empirical price of fairness in failure-aware kidney exchange. *Towards Better and More Affordable Healthcare: Incentives, Game Theory, and Artificial Intelligence* (*HCAGT*) *Workshop at AAMAS-2014* (AAMAS, Richland, SC).

Dickerson JP, Procaccia AD, Sandholm T (2014b) Price of fairness in kidney exchange. *Proc. 2014 Internat. Conf. Autonomous Agents and Multi-Agent Systems* (AAMAS, Richland, SC), 1013–1020.

Dickerson JP, Manlove DF, Plaut B, Sandholm T, Trimble J (2016) Position-indexed formulations for kidney exchange. *Proc. 2016 ACM Conf. Econom. Comput.* (ACM, New York), 25–42.

Edmonds J (1965) Maximum matching and a polyhedron with 0, 1 vertices. *J. Res. Nat. Bur. Standards* 69B(1–2):125–130.

Erdős P, Rényi A (1960) On the evolution of random graphs. *Publications Math. Inst. Hungarian Acad. Sci.* 5:17–61.

Gentry SE, Segev DL (2011) The honeymoon phase and studies of nonsimultaneous chains in kidney-paired donation. *Amer. J. Transplantation* 11(12):2778–2779.

Gentry SE, Montgomery RA, Swihart BJ, Segev DL (2009) The roles of dominos and nonsimultaneous chains in kidney paired donation. *Amer. J. Transplantation* 9(6):1330–1336.

Glorie KM (2012) Estimating the probability of positive crossmatch after negative virtual crossmatch. Technical report, Erasmus School of Economics, Rotterdam, Netherlands.

Glorie KM, van de Klundert JJ, Wagelmans APM (2014) Kidney exchange with long chains: An efficient pricing algorithm for clearing barter exchanges with branch-and-price. *Manufacturing Service Oper. Management* 16(4):498–512.

Goel G, Tripathi P (2012) Matching with our eyes closed. *Sympos. Foundations Comput. Sci.* (*FOCS*) (IEEE, New York), 718–727.

Hooker JN, Williams HP (2012) Combining equity and utilitarianism in a mathematical programming model. *Management Sci.* 58(9):1682–1693.

IBM ILOG Inc. (2010) CPLEX 12.2 User's Manual.

Janson S, Luczak T, Rucinski A (2011) *Random Graphs*, Wiley Series in Discrete Mathematics and Optimization (John Wiley & Sons, New York).

Kidney Paired Donation Work Group (2012) OPTN KPD pilot program cumulative match report (CMR) for KPD match runs: October 27, 2010–November 12, 2012.

Kidney Paired Donation Work Group (2013) OPTN KPD pilot program cumulative match report (CMR) for KPD match runs: October 27, 2010–April 15, 2013.

Leider S, Roth AE (2010) Kidneys for sale: Who disapproves, and why? *Amer. J. Transplantation* 10(5):1221–1227.

Leishman R, Formica R, Andreoni K, Friedewald J, Sleeman E, Monstello C, Stewart D, Sandholm T (2013) The organ procurement and transplantation network (OPTN) kidney paired donation pilot program (KPDPP): Review of current results. *Amer. Transplant Congress* (*ATC*). Talk abstract.

Li J, Liu Y, Huang L, Tang P (2014) Egalitarian pairwise kidney exchange: Fast algorithms via linear programming and parametric flow. *Proc. 2014 Internat. Conf. Autonomous Agents and Multi-Agent Systems* (AAMAS, Richland, SC), 445–452.

Li Y, Kalbfleisch J, Song PX, Zhou Y, Leichtman A, Rees M (2011) Optimization and simulation of an evolving kidney paired donation (KPD) program. Department of Biostatistics Working Paper 90, University of Michigan, Ann Arbor.

Liu Y, Tang P, Fang W (2014) Internally stable matchings and exchanges. *Twenty-Eighth AAAI Conf. Artificial Intelligence* (AAAI, Menlo Park, CA), 1433–1439.

Manlove D, O'Malley G (2015) Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *ACM J. Experiment. Algorithmics* 19(1).

Molinaro M, Ravi R (2011) The query-commit problem, http://arxiv.org/abs/1110.0990.

Montgomery R, Gentry S, Marks WH, Warren DS, Hiller J, Houp J, Zachary AA, et al. (2006) Domino paired kidney donation: A strategy to make best use of live nondirected donation. *The Lancet* 368(9533):419–421.

Park K, Moon JI, Kim SI, Kim YS (1999) Exchange donor program in kidney transplantation. *Transplantation* 67(2):336–338.

Rapaport FT (1986) The case for a living emotionally related international kidney donor exchange registry. *Transplantation Proc.* 18(3, suppl. 2):5–9.

Rees M, Kopke J, Pelletier R, Segev D, Rutter M, Fabrega A, Rogers J, et al. (2009) A nonsimultaneous, extended, altruistic-donor chain. *New England J. Medicine* 360(11):1096–1101.

Ross L, Rubin D, Siegler M, Josephson M, Thistlethwaite J, Woodle S (1997) Ethics of a paired-kidney-exchange program. *New England J. Medicine* 336(24):1752–1755.

Roth A (2007) Repugnance as a constraint on markets. *J. Econom. Perspect.* 21(3):37–58.

Roth A, Sönmez T, Ünver U (2004) Kidney exchange. *Quart. J. Econom.* 119(2):457–488.

Roth A, Sönmez T, Ünver U (2005a) A kidney exchange clearinghouse in New England. *Amer. Econom. Rev.* 95(2):376–380.

Roth A, Sönmez T, Ünver U (2005b) Pairwise kidney exchange. *J. Econom. Theory* 125(2):151–188.

Roth A, Sönmez T, Ünver U (2007) Efficient kidney exchange: Coincidence of wants in a market with compatibility-based preferences. *Amer. Econom. Rev.* 97(3):828–851.

Roth A, Sönmez T, Ünver U, Delmonico F, Saidman SL (2006) Utilizing list exchange and nondirected donation through "chain" paired kidney donations. *Amer. J. Transplantation* 6(11): 2694–2705.

Saidman SL, Roth A, Sönmez T, Ünver U, Delmonico F (2006) Increasing the opportunity of live kidney donation by matching for two and three way exchanges. *Transplantation* 81(5): 773–782.

Sönmez T, Ünver MU (2013) Market design for kidney exchange. Vulkan N, Roth AE, Neeman Z, eds. *The Handbook of Market Design* (Oxford University Press, Oxford, UK), 93–137.

Sönmez T, Ünver MU (2014) Altruistically unbalanced kidney exchange. *J. Econom. Theory* 152(1):105–129.

Stewart D, Leishman R, Sleeman E, Monstello C, Lunsford G, Maghirang J, Sandholm T, et al. (2013) Tuning the OPTN's KPD optimization algorithm to incentivize centers to enter their easy-to-match pairs. *Amer. Transplant Congress* (*ATC*). Talk abstract.

Toulis P, Parkes DC (2015) Design and analysis of multi-hospital kidney exchange mechanisms using random graphs. *Games Econom. Behav.* 91(May):360–382.

Ünver U (2010) Dynamic kidney exchange. *Rev. Econom. Stud.* 77(1):372–414.

Yılmaz Ö (2011) Kidney exchange: An egalitarian mechanism. *J. Econom. Theory* 146(2):592–618.