Failure-Aware Kidney Exchange
(Extended version of an EC-13 paper)

John P. Dickerson, Ariel D. Procaccia, Tuomas Sandholm

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213

{dickerson@cs.cmu.edu, arielpro@cs.cmu.edu, sandholm@cs.cmu.edu}

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 2013; (iii) on synthetic data generated via a model of dynamic kidney exchange. 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 experimentally that taking failed parts from an initial match and instantaneously *rematching* them with other vertices still in the waiting pool can result in significant gains.

Key words: kidney exchange, stochastic matching, discounted cycle cover, random graphs

1. Introduction

Kidney exchange is a recent innovation that allows patients who suffer from terminal kidney failure, and have been lucky 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, in 2012 34,837 people joined the national waiting list while only 10,851 left it due to receiving a kidney. With a median waiting time ranging from 2 to 5 years, 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 131 transplant centers and now conducts weekly match runs. Based on this experience, we propose a significantly different approach as a solution to what we view as the main problem

1

kidney exchanges face today: "last-minute" failures.[2] We mean failures before the transplant procedure 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)). 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 generator and simulator to date, and with the real data from all the UNOS match runs conducted so far.

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 *discounted 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 discounted 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 the donor kidney (or item, in general) of 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 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 US, 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, due to 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-discounted) utility of chains and cycles.

2.2.1. Discounted 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 *discounted 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. Discounted 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, (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, e.g., 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 the 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. Discounted 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 discounted 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 *discounted 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-discounted) 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 Sections 7 and 8.

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 discounted matching that has linearly higher 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) (§4.2), which is an adaptation of the standard *theoretical* kidney exchange model to pools with highly and non-highly 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.

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

(a) A Y gadget. (b) The maximum cardinality matching M_Y. (c) The matching M'_Y.
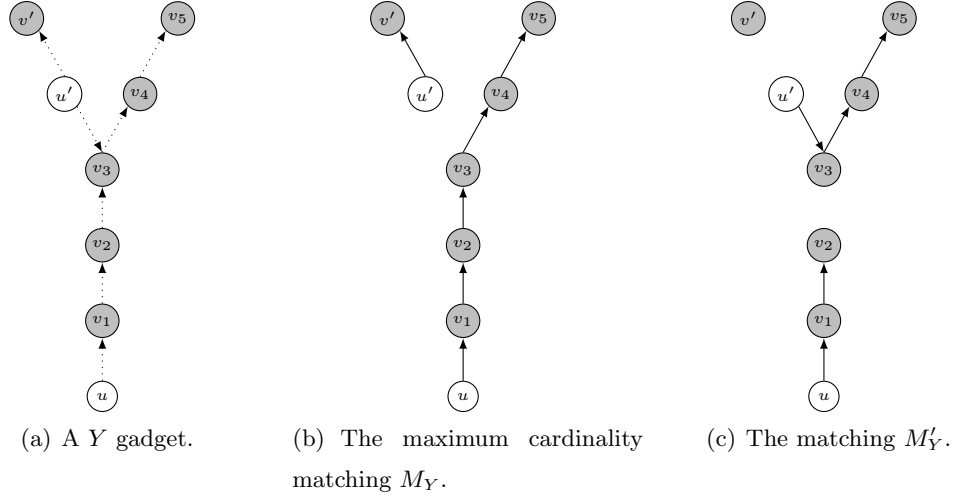
3.2. Maximum cardinality matching in highly sensitized pools

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. By "almost surely" we mean that the probability goes to 1 as n goes to infinity.

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 $G(n, \alpha n, \beta/n)$, there almost surely 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 2(a) for an illustration.

We first claim that it is sufficient to demonstrate that the graph $G(n, \Theta(n), \Theta(1/n))$ almost surely 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 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 2(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 2(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,

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

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 random graphs (see, e.g., Janson et al. (2011)) states that for every graph X of constant size, we can almost surely find $\Omega(n)$ subgraphs that are *isomorphic* to X and *isolated* from the rest of the graph. In particular, our random graph almost surely has $\Omega(n)$ subgraphs that are isomorphic to the undirected, unlabeled version of a Y-gadget.

There are two 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 $\alpha \le 1$. 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. Because we observed far less 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 $r \cdot \min \left\{ cn, \frac{\alpha n}{10k} \right\}$, i.e., $\Omega(n)$. Using Chernoff's inequality, we can almost surely find $\Omega(n)$ correctly labeled gadgets.

We next address the second issue, that is, 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

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

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, a constant fraction of the correctly labeled gadgets will almost surely have correctly oriented edges.
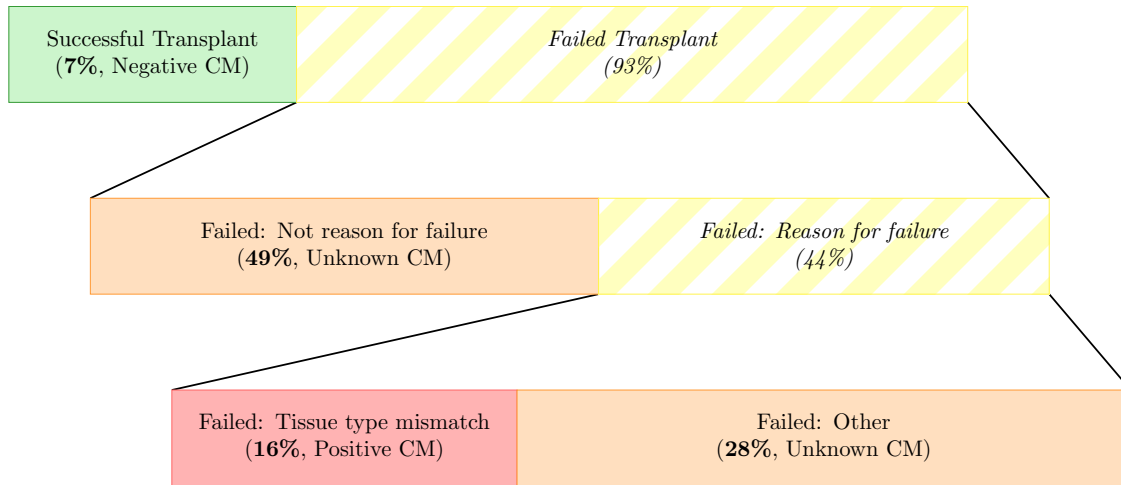
A possible concern at this point is that there are no edges between pairs of vertices in A, and the probability of edges (u, v) where $u \in A$ and $v \in P$ is smaller than p. We first note that, since we are looking at a denser graph, isolation is harder to achieve. Moreover, since a Y-gadget has no 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) 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 June 2013, it matches on a weekly basis, and interacts nationwide with 131 hospitals. Sandholm and Dickerson 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 discounted and non-discounted matching on real data from this exchange, using multiple estimated distributions over edge failure probabilities.

Figure 2   Determining the probability of a match failing is difficult because many potential patient-donor pairs are not crossmatched. Of the aggregate UNOS data, we are only sure that the 7% who successfully received a transplant and the 16% who explicitly failed due to a positive crossmatch were tested.

Successful Transplant (**7%**, Negative CM)	*Failed Transplant (93%)*

Failed: Not reason for failure (**49%**, Unknown CM)	*Failed: Reason for failure (44%)*

Failed: Tissue type mismatch (**16%**, Positive CM)	Failed: Other (**28%**, Unknown CM)

4.1. Estimating edge failure probabilities

The UNOS US-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 takes into account such factors as donor and patient location, health, and CPRA score. We have access to this data, and use it 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 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 due to a *positive crossmatch*, signifying blood-type incompatibility (beyond the ABO model).

—64% failed due to 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 Section 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. 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]

4.2. UNOS results: Discounted matching is better in practice

We now simulate probabilistic matching on real data from UNOS. 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 discounted matching on each real UNOS matching. On the latter, we simulated failure probabilities at least 300 times for each UNOS match run.

Figure 3 shows that, in both cases, taking failure probabilities into account results in significantly more expected transplants. In the constant probability case, discounted 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 discounted 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 US), as discussed by Dickerson et al. (2012b) and Ashlagi et al. (2012).

In the bimodal case, discounted matching shines, often beating the maximum cardinality matching by a factor between 2 and 10, and again never doing worse in expectation. Here, the discounted

Figure 3   Comparison of the expected number of transplants resulting from the maximum weighted matching and discounted weighted matching methods, on 51 UNOS match runs between October 2010 and May 2013. (Left) Constant edge success probability. (Right) Bimodal edge success probabilities (some very high, some very low). In both cases, taking edge failures into account results in equal or better expected transplant counts (often much better).
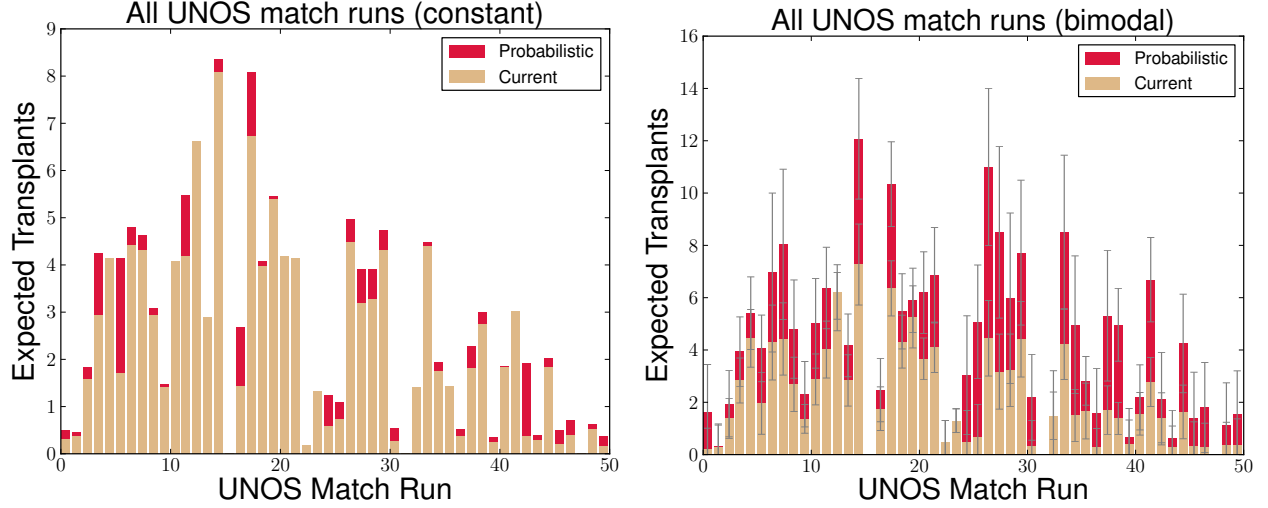

Table 1   Distributional difference between maximum weighted matching and failure-aware matching on real UNOS data.

Distribution	Current		Ours		t-test		Wilcoxon Signed-rank	
	Average	St. Dev.	Average	St. Dev	t-statistic	p-value	Effect Size	p-value
Constant	2.30	2.24	2.63	2.36	4.61	$< 10^{-4}$	8.12	$< 10^{-6}$
Bimodal	2.80	2.08	5.97	4.25	8.23	$< 10^{-10}$	0.98	$< 10^{-6}$

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 probability distributions. Across all UNOS match runs using a constant edge failure probability of 0.7, the failure-aware method results in an expected 0.33 extra transplants per match run than maximum weighted solver. Using the bimodal distribution, the failure-aware method returns an expected additional 3.17 transplants per match run. Both results 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.
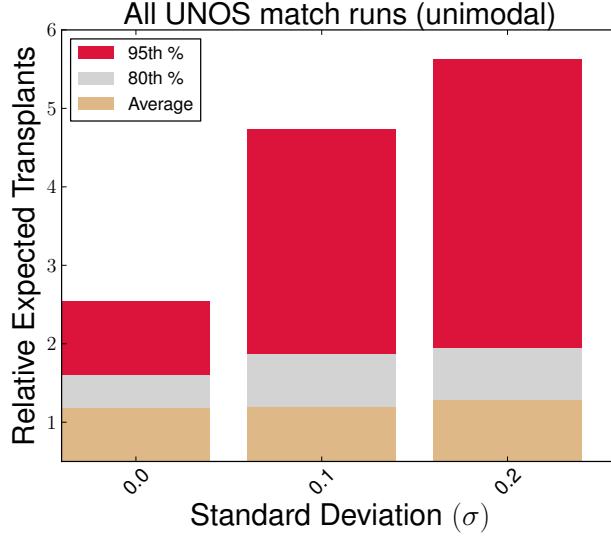
Figure 4 Aggregate additional transplants over all UNOS match runs, for edge failure probabilities drawn randomly from $\mathcal{N}(\mu = 0.7, \sigma \in \{0.1, 0.2, 0.35, 0.5\})$. The leftmost point "$\sigma = 0.0$" represents a constant failure rate of 70%.

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 4 shows the aggregate number of expected transplants (summed over all UNOS match runs) 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, 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).

Next, in Section 5, we construct a solver that can optimally solve capable of clearing large exchanges than those currently available at UNOS.

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 186 patients and 205 donors. However, as kidney exchange gains traction, these pools will grow. As discussed by Abraham et al. (2007), the estimated steady-state size of a US nationwide kidney exchange is 10,000 patients.

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

PROPOSITION 1. *The discounted 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 for each cycle no longer than L (in practice, $L = 3$) and constraints that state that accepted cycles are vertex disjoint. 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. Due to 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 undiscounted case (as shown by Dickerson et al. (2012b)).

In this section, we augment the current UNOS solver to solve the discounted 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 discounted 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 discounted version of this problem, Proposition 2 shows that computing this bound is NP-hard.

PROPOSITION 2. *The unrestricted discounted 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 the cycles in the constructed widgets in their proof are of length at least 3. Due to discounting, a perfect cover which 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 $k > 3$ due to discounting. 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 uses a looser upper bound, solving the unrestricted clearing problem on a graph $G' = (V, E')$ such that $w'_e = w_e q_e$, for each $e \in E$.

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 undiscounted clearing problem for $L = 2$) as a primal heuristic, or lower bound. The new solver uses the *discounted* 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 discounted 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 two-cycle between those vertices in G. Then, set the weight of every edge $e' = (v_i, v_j)$ in G' to:

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

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 due to Saidman et al. (2006), pools of size 5000 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 discounted 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:

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

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 $V' = V \setminus c$). 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:

$$u(c^\infty) - u(c) =_{k \to \infty} \frac{q_{max}}{1 - q_{max}} \prod_{i=0}^{k-1} q_i \tag{2}$$

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 (2) 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:

$$\left(\frac{q_{max}}{1 - q_{max}} \prod_{i=0}^{k-1} q_i \right) + u(c) + \ell - \left(d_{min} + \sum_{i=0}^{k} d_i \right) \leq 0 \tag{3}$$

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 3 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 3 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 3 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 discounted 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 non-altruist 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 non-positive 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 non-positive 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 $\nu_{a,v} \leq 0$. Due to 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.

Table 2   Scaling results for our method versus CPLEX, timeout of 1 hour.

	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	–
1000	0 / 128	–	0 / 128	–	0 / 128	–

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 2 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 8GB 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 (§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 (§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 (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 discounted 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 2 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. Instantaneous rematching in the static model

Fielded kidney exchanges operate in the *static* setting, first performing a batch matching (typically at a defined periodicity), then testing edges in that match, and finally performing successfully tested transplants and placing patient-donor pairs with failed edges back in the pool. In this section, we explore the effect of this policy in the static setting (which leads to Section 8 and the formulation of a general dynamic model of kidney exchange).

A myopic optimization (using the failure-aware method described in this paper) is performed without taking into account the possibility of rematching. The leftover pool—together with the positive crossmatches from the last match—are then instantaneously rematched. We are interested in the additional expected transplants gained from this second (or third, or more) round of matching. Note that this only makes sense in a model that includes match failures, as the maximum weighted matching in a deterministic setting would leave no matchable vertices in the remaining pool.

Figure 5 shows the effect of instantaneous rematching on both the size of the discounted maximum matching and the expected number of resulting transplants, as the number of allowed rematches is increased from zero (a single batch match) to nine (ten total matches, nine instantaneous rematches). While the size and value of subsequent matches decreases (as expected), the

Figure 5   Expected cardinality of the algorithmic match (dotted line) and number of transplants (solid line) as
the number of simultaneous rematches is increased, for $|V| \in \{50, 250, 500\}$ and $|A| = 0.1|V|$.
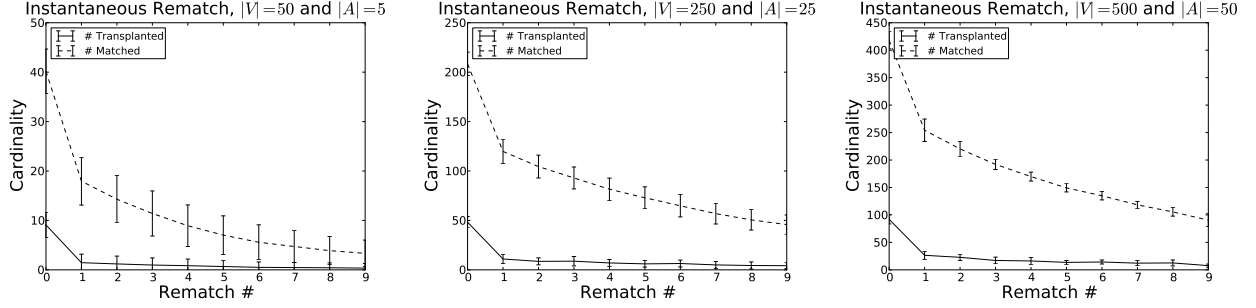

Table 3   Expected number of transplants given a single matching ("1M")
versus a single matching and nine rematches ("10M").

$|V|$	1M Avg. (St. Dev.)	10M Avg. (St. Dev.)	Percent Change
50	9.09 (2.56)	15.97 (3.98)	+75.69%
100	18.25 (3.61)	36.95 (6.35)	+102.02%
250	48.23 (5.69)	109.66 (9.84)	+127.27%
500	92.00 (8.28)	235.00 (16.25)	+155.43%

results have a heavy tail; that is, even later rounds of rematching contribute nontrivially to the
aggregate expected transplants in large enough pools.

Table 3 quantifies the heavy tails shown in Figure 5. It compares the expected number of trans-
plants after a single batch matching against the aggregate expected value of a single batch matching
and nine instantaneous rematches. Even in a small pool with 50 patient-donor pairs and 5 altruists,
multiple rematches result in an expected additional 75.69% transplants. Greater percentage-wise
gains are realized for pools of larger size due to the remaining thickness in an exchange, even after
multiple vertex and edge removals.

8. 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 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 4 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 due to, e.g., crossmatch failures or donor refusals. Vertices are removed if the patient or
her respective donor must leave the pool, due to reasons ranging from a successful transplantation
to patient expiration.

Table 4   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**	
Altruist runs out of patience		
Bridge donor reneges		

*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.

Figure 6   The evolution dynamics of a kidney exchange.
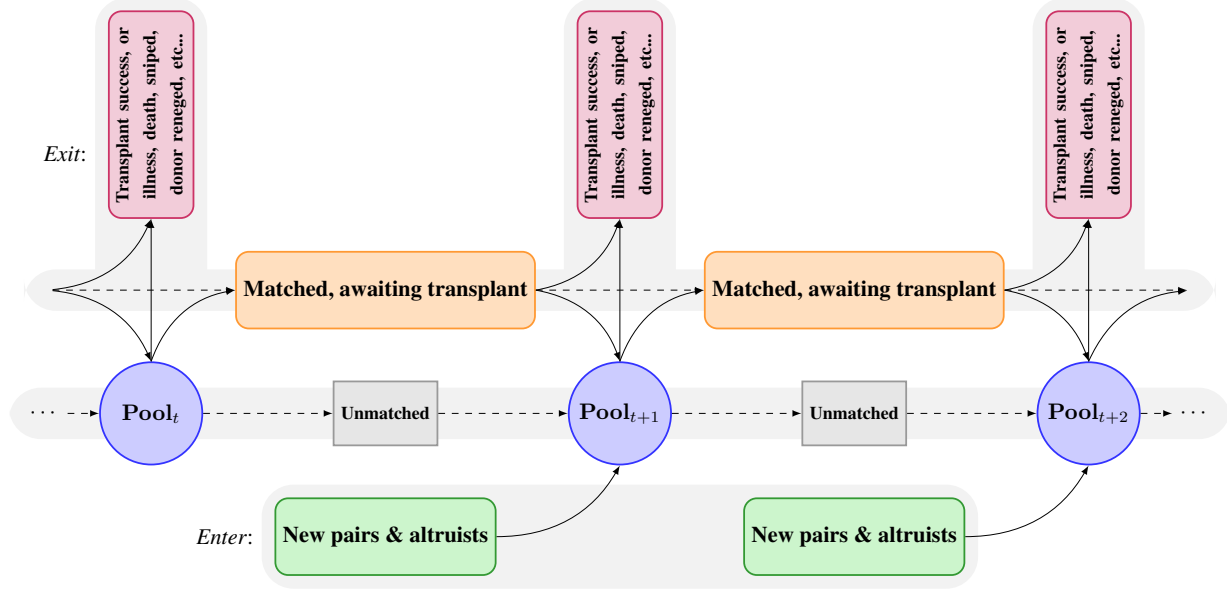
Figure 6 provides a snapshot of a compatibility graph over three points in time. The pool at time t consists of unmatched patients and donors from time $t-1$, any new pairs and altruists entering the pool, and any vertices who were waiting for a successful match, but whose match failed (due to, e.g., 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 4, or are temporarily marked "inactive" through a pending match.

8.1. 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

Figure 7   Expected number of transplants per week for graphs of different sizes. From left to right, 5 pairs and 1 altruist, 20 pairs and 4 altruists, and 25 pairs and 5 altruists (on expectation) appear every week.
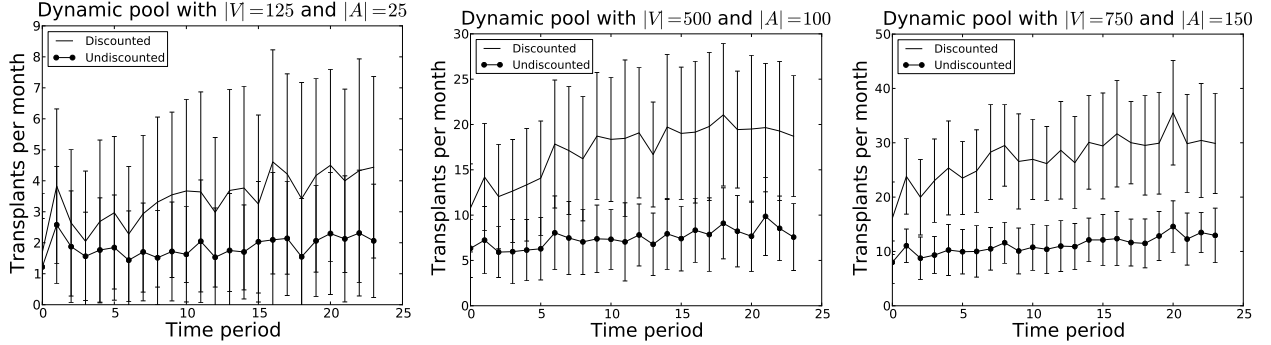

represents current kidney exchanges. The deceased-donor waitlist donation at the end of a chain is counted in the expected number of transplants.

In our experience with UNOS, typically the time between a match offer and transplant success or failure 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—or is removed permanently, if the reason for failure was that patient-donor pair's exit from the exchange.

Figure 7 shows the number of expected transplants per week on graphs of three different sizes. In expectation, 5, 20, or 25 pairs and 1, 4, or 5 altruists appear weekly in each of the three graphs. Discounted 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 gives aggregate results for total number of expected transplants over 24 weeks, for graphs of varying size, for both discounted and maximum cardinality matching. Graphs have 10% as many altruists on top of the patient-donor pool. The gap between discounted and non-discounted matching widens as the activity level of the dynamic kidney exchange increases. For our largest graphs, discounted matching improved expected transplants by a factor of three over maximum cardinality matching.

Figure 8   Expected aggregate transplants over 24 weeks, for increasing $|V|$ (and $|A| = 0.1|V|$).
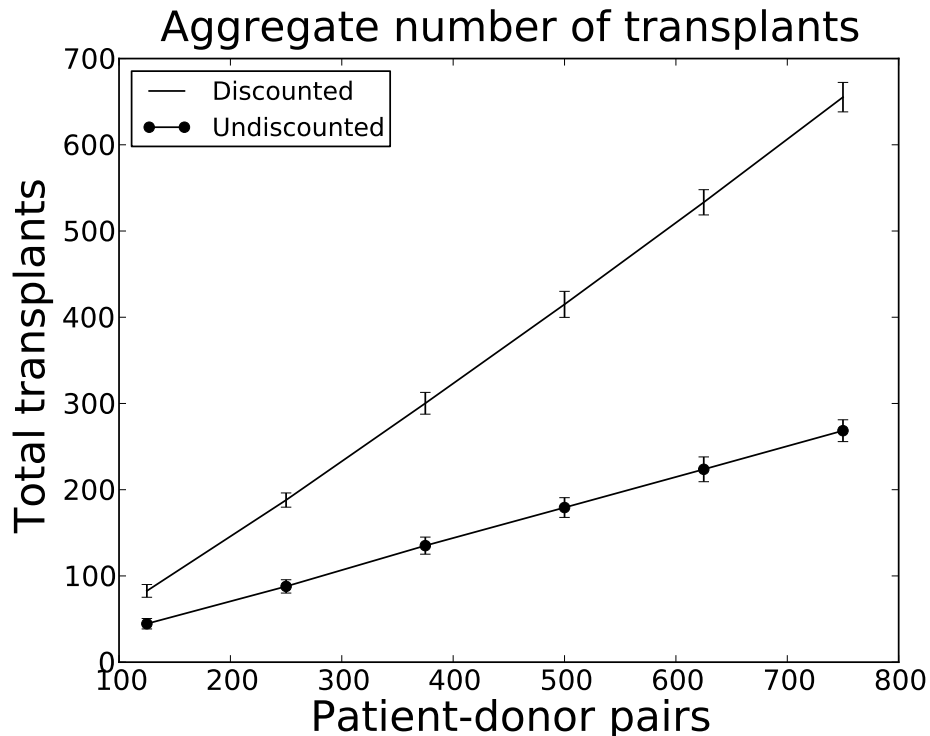
9. Additional related work

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

Analysis of kidney exchange using random graph models is the *de facto* method for providing theoretical guidance. We work with the model of Ashlagi et al. (2012); related random models include those of Ashlagi and Roth (2011) and Toulis and Parkes (2011).

The work on the query-commit problem by Molinaro and Ravi (2013) 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 *two* crossmatches 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 this model.

10. 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 non-maximum 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 2013, and found that our failure-aware matching increases the number of expected transplants dramatically.

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 also explored the idea of immediately reentering failed cycles and chain segments from an initial matching back into the waiting pool and subsequently rerunning the matching algorithm again; this instantaneous rematching results in significant extra transplants and can be performed multiple times with relative ease. We then developed a faithful 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.

Experimentally, our solver would benefit from a better (i.e., tighter) upper bound on the discounted clearing problem—the current bound is especially loose when failure probabilities are high and when bounding the utility of long cycles and long chains. Tightening the upper bound would decrease the size of the search tree and, in turn, reduce column generation and overall runtime.

The accuracy of our and others' experimental results relative to real kidney exchange will continue to improve as we work with more exchanges. The community's understanding of the underlying failure probabilities—especially on a patient-by-patient basis—will improve as more data becomes available. Quantitatively analyzing this data would be of great help to both science and practice of kidney exchange.

Striking a balance between fairness and efficiency in kidney exchange is an increasingly important line of work combining medical policy, economics, and optimization. Roth, Sömnez, and Ünver (2005) 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, in fact, performed in the current UNOS exchange). Recent work by Bertsimas, Farias, and Trichakis (2011, 2012) studies tradeoffs between fairness and efficiency in general resource allocation problems, and may be applicable to kidney exchange. Bertsimas, Farias, and Trichakis (2013) then 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.

This paper motivates a move to discounted kidney exchange optimization from an 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 discounted approach would disfavor highly-sensitized patients. Yet, data from the UNTOS kidney exchange (reported by the Kidney Paired Donation Work Group (2013)) does not show a correlation between post-match failure and CPRA. This 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 due to 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. By lowering these non-medical reasons for failure (e.g., by merging all exchanges into a single program to reduce inter-exchange "sniping"), the overall failure rate for highly-sensitized patients would probably become higher than that of other patients. If that turns out to be the case, further research should be conducted on the efficiency-fairness tradeoff in the context of discounted kidney exchange optimization. As an extreme approach, one could simply set all the probabilities in the optimization to be equal in order to not disfavor patients with high failure probabilities. Even with this extreme approach, the

discounted framework would strike good endogenous tradeoffs between short chains, long chains, short cycles, and long cycles—unlike the current undiscounted approach.

Endnotes

1. This paper is a direct but significant extension of Dickerson et al. (2013a), and includes new computational ideas, techniques, and results on real kidney exchange data from the UNOS nation-wide kidney exchange from its inception in 2010 through May of 2013 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 American Transplant Congress (especially regarding work by Leishman et al. (2013) and Dickerson et al. (2013b)).

2. Another challenge in kidney exchanges is that transplant centers may be motivated to hide some of their donor-patient pairs and altruistic donors. 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 (2011), Ashlagi et al. (2010)). So, 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 is now available in a report from the Kidney Paired Donation Work Group (2013); this most recent data was not incorporated into our experiments, but is very similar to that which was.

4. We explore a wide variety of underlying edge failure probability distributions for our UNOS experiments, in part because the underlying real-world distribution is not known. 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. Yet, data from the UNOS kidney exchange (reported by the Kidney Paired Donation Work Group (2013)) does not show a correlation between post-match failure and CPRA. This 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 due to 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. By lowering these non-medical reasons for failure (e.g., by merging all exchanges into a single program to reduce inter-exchange "sniping"), the overall failure rate for highly-sensitized patients would probably become higher than that of other patients. This would beget increased need for a discussion on equity versus efficiency in kidney exchange. Section 10 surveys the state-of-the-art for this line of research and discusses this issue further.

Acknowledgments

References

Abraham, David, Avrim Blum, Tuomas Sandholm. 2007. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. *ACM Conference on Electronic Commerce (EC)*. 295–304.

Ashlagi, Itai, Felix Fischer, Ian Kash, Ariel D. Procaccia. 2010. Mix and match. *ACM Conference on Electronic Commerce (EC)*. 305–314.

Ashlagi, Itai, David Gamarnik, Michael Rees, Alvin E. Roth. 2012. The need for (long) chains in kidney exchange. NBER Working Paper No. 18202.

Ashlagi, Itai, Duncan S. Gilchrist, Alvin E. Roth, Michael Rees. 2011. Nonsimultaneous chains and dominos in kidney-paired donation—revisited. *American Journal of Transplantation* **11**(5) 984–994.

Ashlagi, Itai, Patrick Jaillet, Vahideh H. Manshadi. 2013. Kidney exchange in dynamic sparse heterogenous pools. *ACM Conference on Electronic Commerce (EC)*. 25–26.

Ashlagi, Itai, Alvin E. Roth. 2011. Individual rationality and participation in large scale, multi-hospital kidney exchange. *ACM Conference on Electronic Commerce (EC)*. 321–322.

Awasthi, Pranjal, Tuomas Sandholm. 2009. Online stochastic optimization in the large: Application to kidney exchange. *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*. 405–411.

Barnhart, Cynthia, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, Pamela H. Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations Research* **46**(3) 316–329.

Bertsimas, Dimitris, Vivek F Farias, Nikolaos Trichakis. 2011. The price of fairness. *Operations Research* **59**(1) 17–31.

Bertsimas, Dimitris, Vivek F Farias, Nikolaos Trichakis. 2012. On the efficiency-fairness trade-off. *Management Science* **58**(12) 2234–2250.

Bertsimas, Dimitris, Vivek F Farias, Nikolaos Trichakis. 2013. Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Operations Research* **61**(1) 73–87.

Blum, Avrim, Anupam Gupta, Ariel D. Procaccia, Ankit Sharma. 2013. Harnessing the power of two crossmatches. *ACM Conference on Electronic Commerce (EC)*. 123–140.

Chen, Yanhua, Yijiang Li, John D. Kalbfleisch, Yan Zhou, Alan Leichtman, Peter X.-K. Song. 2012. Graph-based optimization algorithm and software on kidney exchanges. *IEEE Transactions on Biomedical Engineering* **59** 1985–1991.

Dickerson, John P., Ariel D. Procaccia, Tuomas Sandholm. 2012a. Dynamic matching via weighted myopia with application to kidney exchange. *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. 1340–1346.

Dickerson, John P., Ariel D. Procaccia, Tuomas Sandholm. 2012b. Optimizing kidney exchange with transplant chains: Theory and reality. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 711–718.

Dickerson, John P., Ariel D. Procaccia, Tuomas Sandholm. 2013a. Failure-aware kidney exchange. *ACM Conference on Electronic Commerce (EC)*. 323–340.

Dickerson, John P., Ariel D. Procaccia, Tuomas Sandholm. 2013b. Results about, and algorithms for, robust probabilistic kidney exchange matching. *American Transplant Congress (ATC)*. Poster abstract.

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

Erdős, Paul, Alfréd Rényi. 1960. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* **5** 17–61.

Gentry, Sommer E., Robert A. Montgomery, Bruce J. Swihart, Dorry L. Segev. 2009. The roles of dominos and nonsimultaneous chains in kidney paired donation. *American Journal of Transplantation* **9**(6) 1330–1336.

Gentry, Sommer E., Dorry L. Segev. 2011. The honeymoon phase and studies of nonsimultaneous chains in kidney-paired donation. *American Journal of Transplantation* **11**(12) 2778–2779. doi:10.1111/j.1600-6143.2011.03802.x.

Goel, Gagan, Pushkar Tripathi. 2012. Matching with our eyes closed. *Symposium on the Foundations of Computer Science (FOCS)*. IEEE, 718–727.

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

Janson, Svante, Tomasz Luczak, Andrzej Rucinski. 2011. *Random Graphs*. Wiley Series in Discrete Mathematics and Optimization, Wiley.

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

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

Leishman, Ruthanne, Richard Formica, Kenneth Andreoni, John Friedewald, Elizabeth Sleeman, Catherine Monstello, Darren Stewart, Tuomas Sandholm. 2013. The Organ Procurement and Transplantation Network (OPTN) Kidney Paired Donation Pilot Program (KPDPP): Review of current results. *American Transplant Congress (ATC)*. Talk abstract.

Li, Yijiang, Jack Kalbfleisch, Peter Xuekun Song, Yan Zhou, Alan Leichtman, Michael Rees. 2011. Optimization and simulation of an evolving kidney paired donation (KPD) program. Department of biostatistics working paper 90, University of Michigan.

Molinaro, Marco, R. Ravi. 2013. Kidney exchanges and the query-commit problem. Manuscript.

Montgomery, Robert, Sommer Gentry, William H Marks, Daniel S Warren, Janet Hiller, Julie Houp, Andrea A Zachary, J Keith Melancon, Warren R Maley, Hamid Rabb, Christopher Simpkins, Dorry L Segev. 2006. Domino paired kidney donation: a strategy to make best use of live non-directed donation. *The Lancet* **368**(9533) 419–421.

Rees, Michael, Jonathan Kopke, Ronald Pelletier, Dorry Segev, Matthew Rutter, Alfredo Fabrega, Jeffrey Rogers, Oleh Pankewycz, Janet Hiller, Alvin Roth, Tuomas Sandholm, Utku Ünver, Robert Montgomery. 2009. A nonsimultaneous, extended, altruistic-donor chain. *New England Journal of Medicine* **360**(11) 1096–1101.

Roth, Alvin, Tayfun Sönmez, Utku Ünver, Frank Delmonico, Susan L. Saidman. 2006. Utilizing list exchange and nondirected donation through 'chain' paired kidney donations. *American Journal of Transplantation* **6** 2694–2705.

Roth, Alvin E, Tayfun Sömnez, Utku Ünver. 2005. Pairwise kidney exchange. *Journal of Economic Theory* **125**(2) 151–188.

Saidman, Susan L., Alvin Roth, Tayfun Sönmez, Utku Ünver, Frank Delmonico. 2006. Increasing the opportunity of live kidney donation by matching for two and three way exchanges. *Transplantation* **81**(5) 773–782.

Stewart, Darren, Ruthanne Leishman, Elizabeth Sleeman, Catherine Monstello, Guy Lunsford, Jude Maghirang, Tuomas Sandholm, Sommer Gentry, Richard Formica, John Friedewald, Kenneth Andreoni. 2013. Tuning the OPTN's KPD optimization algorithm to incentivize centers to enter their easy-to-match pairs. *American Transplant Congress (ATC)*. Talk abstract.

Toulis, Panos, David C. Parkes. 2011. A random graph model of kidney exchanges: efficiency, individual-rationality and incentives. *ACM Conference on Electronic Commerce (EC)*. ACM, 323–332.

Ünver, Utku. 2010. Dynamic kidney exchange. *Review of Economic Studies* **77**(1) 372–414.