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

OPERATIONS RESEARCH
Vol. 68, No. 1, January–February 2020, pp. 16–34
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Crosscutting Areas

Ignorance Is Almost Bliss: Near-Optimal Stochastic Matching with Few Queries

Avrim Blum,[a] John P. Dickerson,[b] Nika Haghtalab,[c] Ariel D. Procaccia,[d] Tuomas Sandholm,[d] Ankit Sharma[d]

[a] Toyota Technological Institute, Chicago, Illinois 60637; [b] University of Maryland, College Park, Maryland 20742; [c] Microsoft Research, Cambridge, Massachusetts 02142; [d] Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

Contact: avrim@ttic.edu, http://orcid.org/0000-0003-2450-5102 (AB); john@cs.umd.edu, http://orcid.org/0000-0003-2231-680X (JPD); nika.haghtalab@microsoft.com, http://orcid.org/0000-0002-8612-2089 (NH); arielpro@cs.cmu.edu, http://orcid.org/0000-0003-0318-491X (ADP); sandholm@cs.cmu.edu, http://orcid.org/0000-0001-8861-9366 (TS); ankits@cs.cmu.edu, http://orcid.org/0000-0002-5646-5929 (AS)

Received: December 21, 2016
Revised: July 25, 2018
Accepted: December 13, 2018
Published Online in Articles in Advance: January 6, 2020

Subject Classifications: networks/graphs: matching, stochastic, theory
Area of Review: Games, Information, and Networks

https://doi.org/10.1287/opre.2019.1856

Abstract. We study the stochastic matching problem with the goal of finding a maximum matching in a graph whose edges are unknown but can be accessed via queries. This is a special case of stochastic k-cycle packing, in which the problem is to find a maximum packing of cycles, each of which exists with some probability. We provide polynomial-time *adaptive* and *nonadaptive* algorithms that provably yield a near-optimal solution, using a number of edge queries that is linear in the number of vertices. We are especially interested in kidney exchange, with which pairs of patients with end-stage renal failure and their willing but incompatible donors participate in a mechanism that performs *compatibility tests* between patients and donors and swaps the donors of some patients so that a large number of patients receive compatible kidneys. Because of the significant cost of performing compatibility tests, currently, kidney exchange programs perform at most one compatibility test per patient. Our theoretical results applied to kidney exchange show that, by increasing the number of compatibility tests performed per patient from one to a larger constant, we effectively get the full benefit of exhaustive testing at a fraction of the cost. We show, on both generated and real data from the UNOS nationwide kidney exchange, that even a small number of nonadaptive edge queries per vertex results in large gains in expected successful matches.

Funding: This work was supported by the National Science Foundation Division of Computing and Communication Foundations [Grants 1101668, 1116892, 1215883, 1415460, 1525971, and 1535967] and Division of Information and Intelligent Systems [Grants 0964579, 1065251, 1320620, 1350598, 1617590, and 1546752], the Army Research Office [W911NF-16-1-0061], a National Defense Science & Engineering Graduate Fellowship, an Alfred P. Sloan Foundation Research Fellowship, and a Microsoft Research PhD Fellowship.
Supplemental Material: The e-companion is available at https://doi.org/10.1287/opre.2019.1856.

Keywords: stochastic matching • kidney exchange • matching with queries

1. Introduction

In the *stochastic matching* problem, we are given an undirected graph $G = (V, E)$, in which we do not know which edges in E actually exist. Rather, for each edge $e \in E$, we are given an existence probability p_e. Of interest, then, are algorithms that first query some subset of edges to find ones that exist and, based on these queries, produce a matching that is as large as possible. The stochastic matching problem is a special case of *stochastic k-cycle packing*, in which each cycle exists only when all of its edges exist, and the goal is to find a (vertex disjoint) packing of existing cycles that collectively cover the maximum number of vertices possible.

Without any constraints, one can simply query all edges and then output the maximum matching or packing over those that exist—hereafter, referred to as the omniscient optimal solution. But this level of freedom may not always be available; therefore, we are interested in the trade-off between the number of queries and the fraction of the omniscient optimal solution achieved. Specifically, we ask: to perform as well as the omniscient optimum in the stochastic matching problem, do we need to query (almost) all the edges; that is, do we need a budget of $\Theta(n)$ queries per vertex, where n is the number of vertices? Or can we, for any arbitrarily small $\epsilon > 0$, achieve a $(1 - \epsilon)$ fraction of the omniscient optimum by using an $o(n)$ per-vertex budget? We answer these questions as well as their extensions to the k-cycle packing problem. We support our theoretical results empirically on both generated and real data from a large-fielded kidney exchange in the United States.

1.1. Our Theoretical Results and Techniques

Our main theoretical result gives a positive answer to the latter question for stochastic matching by

showing that, surprisingly, a *constant* per-vertex budget is sufficient to get ϵ-close to the omniscient optimum. Indeed, we design a polynomial-time algorithm with the following properties: for any constant $\epsilon > 0$, the algorithm queries at most $O_\epsilon(1)$ edges incident to any particular vertex, requires $O_\epsilon(1)$ rounds of parallel queries, and achieves $(1-\epsilon)$ fraction of the omniscient optimum. This guarantee holds as long as all the nonzero p_e's are bounded away from zero by some constant that is independent of n (see Section 9 for a discussion of cases in which p_e may be arbitrarily small for a few edges). Notation $O_\epsilon(1)$ refers to asymptotic behavior that is constant when ϵ is a fixed constant. When it is clear from the context, we use $O(1)$ instead of $O_\epsilon(1)$.

The foregoing algorithm is *adaptive* in the sense that its queries are conditioned on the answers to previous queries. Even though it requires only a constant number of rounds, it is natural to ask whether a nonadaptive algorithm—one that issues all its queries in one round—can also achieve a similar guarantee. We do not give a complete answer to this question, but we do present a nonadaptive algorithm that achieves a $0.5(1-\epsilon)$-approximation (for arbitrarily small $\epsilon > 0$) to the omniscient optimum. In Appendix EC.1, we extend our matching results to a more general stochastic model, in which the probability of existence of an edge depends on parameters associated with its two end points.

We also extend our results to the stochastic k-cycle packing problem, for which we are given a directed graph and the collection of all of its cycles of length at most k. The goal is to find the collection of mutually vertex-disjoint cycles, called a packing, that covers the maximum number of vertices possible. Stochastic matching is a special case of stochastic k-cycle packing: each undirected edge in stochastic matching corresponds to a cycle of length two, that is, $k = 2$. In stochastic k-cycle packing, each cycle exists if and only if all of its edges exist. That is, when p represents the probability of a directed edge existing, then a cycle of length ℓ exists with probability p^ℓ although these events are correlated across cycles that share an edge. Our goal is to query the edges and output a collection of existing *vertex-disjoint* cycles that covers a large number of vertices. We present an adaptive polynomial-time algorithm that, for any constant $\epsilon > 0$, returns a collection of vertex-disjoint cycles that covers a number of vertices that is at least $\frac{4}{k^2}(1-\epsilon)$ of the omniscient optimum using $O_{\epsilon,k}(1)$ queries per element, hence, $O_{\epsilon,k}(n)$ queries overall.

To better appreciate the challenge we face, we note that, even in the stochastic matching setting, we do not have a clear idea of how large the omniscient optimum is. Indeed, there is a significant body of work on the expected cardinality of matching in *complete* random graphs (see, e.g., Bollobás (2001), chapter 7), in which the omniscient optimum is known to be close to n. But, in our work, we are dealing with *arbitrary* graphs in which it can be a much smaller number. In addition, naïve algorithms fail to achieve our goal even if they are allowed many queries. For example, querying a sublinear number of edges incident to each vertex, chosen uniformly at random, gives a vanishing fraction of the omniscient optimum as we show in Section 4.

The primary technical ingredient in the design of our *adaptive algorithm* is that, if, in any round r of the algorithm, the solution computed by round r (based on previous queries) is small compared with the omniscient optimum, then the current structure must admit a *large collection of disjoint constant-sized "augmenting" structures*. These augmenting structures are composed of edges that have not been queried so far. Of course, we do not know whether these structures we are counting on to help augment our current matching actually exist, but we do know that these augmenting structures have constant size (and so each structure exists with some constant probability) and are disjoint (and, therefore, the outcomes of the queries to the different augmenting structures are independent). Hence, by querying all these structures in parallel in round r, in expectation, we can close a constant fraction of the gap between our current solution and the omniscient optimum. By repeating this argument over a constant number of rounds, we achieve a $(1-\epsilon)$ fraction of the omniscient optimum. In the case of stochastic matching, these augmenting structures are simply augmenting paths; in the more general case of k-cycle packing, we borrow the notion of augmenting structures from Hurkens and Schrijver (1989).

1.2. Our Experimental Results: Application to Kidney Exchange

Our work is directly motivated by applications to kidney exchange, a medical approach that enables kidney transplants. Transplanted kidneys are usually harvested from deceased donors, but as of June 10, 2018, there are 114,877 people on the U.S. national waiting list (U.S. Department of Health and Human Services 2018), making the median waiting time dangerously long. Fortunately, kidneys are an unusual organ in that donation by living donors is also a possibility as long as patients happen to be medically compatible with their potential donors.

In its simplest form—*pairwise exchange*—two incompatible donor–patient pairs exchange kidneys: the donor of the first pair donates to the patient of the second pair, and the donor of the second pair donates to the patient of the first pair. This setting can be represented as an undirected *compatibility graph*, in which each vertex represents an incompatible donor–patient pair, and an edge between two vertices represents the

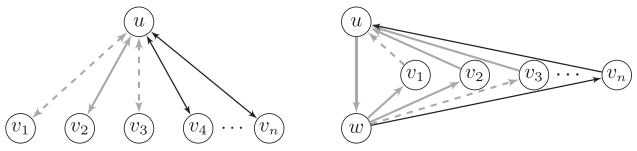
possibility of a pairwise exchange. A matching in this graph specifies which exchanges take place. Modern kidney exchange programs regularly employ swaps involving *three* donor–patient pairs, which are known to provide significant benefit compared with pairwise swaps alone (Roth et al. 2007, Ashlagi and Roth 2014). Mathematically, we can consider a directed graph, in which an edge (u, v) means that the donor of pair u is possibly compatible with the patient of pair v. In this graph, pairwise and three-way exchanges correspond to two-cycles and three-cycles, respectively. See Figure 1 for a demonstration of this model.

The edges of the compatibility graph can be determined based on the medical characteristics—blood type and tissue type—of donors and patients. However, the compatibility graph only tells part of the story. Before a transplant takes place, a more accurate medical test known as a *crossmatch test* and additional consultations with transplant centers take place. This test involves mixing samples of the blood of the patient and the donor (rather than simply looking up information in a database), making the test relatively costly and time-consuming. Consequently, crossmatch tests are only performed for donors and patients that have been matched. Although some patients are more likely to pass crossmatch tests than others—the probability is related to a measure of sensitization known as the person's panel reactive antibody (PRA)—the average is as low as 30% in major kidney exchange programs (Ashlagi et al. 2011b, Leishman et al. 2013, Dickerson et al. 2019). This means that, if we only tested a perfect pairwise matching over n donor–patient pairs, we would expect only $0.09n$ of the patients to actually receive a kidney. In contrast, the omniscient solution that runs crossmatch tests on all possible pairwise exchanges (in the compatibility graph) may be able to provide kidneys to all n patients, but this solution is impractical.

Our adaptive algorithm for stochastic pairwise matching uncovers a sweet spot between these two extremes. On the one hand, it only mildly increases medical expenses from one crossmatch test per patient to a larger, yet constant, number, and it is highly parallelizable, requiring only a constant number of rounds, so the time required to complete all crossmatch tests does not scale with the number of donors and patients. On the other hand, the adaptive algorithm essentially recovers the entire benefit of testing all potentially feasible pairwise exchanges. The qualitative message of this theoretical result is clear: *a mild increase in number of crossmatch tests provides nearly the full benefit of exhaustive testing*. When three-way exchanges are considered, our adaptive algorithm for three-cycle packing provides a (4/9)-approximation to the omniscient optimum, using only $O(1)$ crossmatch tests per patient and $O(n)$ overall. Although the practical implications of this result are currently not as crisp as those of its pairwise counterpart, future work may improve the approximation ratio (using $O(n)$ queries and an exponential-time algorithm) as we explain in Section 9.1.

To bridge the gap between theory and practice, we provide experiments for pairwise and three-way exchanges on both simulated and real data from the first 169 match runs of the United Network for Organ Sharing (UNOS) U.S. nationwide kidney exchange, which now includes 153 transplant centers—approximately 66% of the transplant centers in the United States. The exchange began matching in October 2010 and now matches on a biweekly basis. Using adaptations of the algorithms presented in this paper, we show that even a small number of nonadaptive rounds followed by a single period during which only those edges selected during those rounds are queried results in large gains relative to the omniscient pairwise or three-way exchanges. We discuss the policy implications of this promising result in Section 9.2.

Figure 1. Compatibility Graphs for Pairwise and Three-Way Exchanges



Notes. Solid gray edges represent successful crossmatch tests, dashed gray edges represent failed crossmatch tests, and black edges represent potential compatibilities that have not been tested. Note that, when pairwise exchanges are considered, the number of incoming edge tests of a node is the same as the number of its outgoing edge tests—a patient and its willing but incompatible donor are always involved in an equal number of tests—although in three-way exchanges the number of incoming and outgoing edge tests may be different.

2. Related Work

Although papers on stochastic matching often draw on kidney exchange for motivation—or at least mention it in passing—these two research areas are almost disjoint. We, therefore, discuss them separately in Sections 2.1 and 2.2.

2.1. Stochastic Matching

Prior work has considered multiple variants of stochastic matching. A popular variant is the *query-commit* problem, in which the algorithm is *forced* to add any queried edge to the matching if the edge is found to exist. Goel and Tripathi (2012) establish an upper bound of 0.7916 for graphs in which no information is available about the edges, and Costello et al. (2012) establish a lower bound of 0.573 and an upper bound of 0.898 for graphs in which each edge e exists with a given probability p_e. Molinaro and Ravi (2011) propose an algorithm for two-cycle matching

in the query-commit model that is nearly optimal given additional theoretical assumptions. Similarly to our work, these approximation ratios are with respect to the omniscient optimum, but the informational disadvantage of the algorithm stems purely from the query-commit restriction.

Within the query-commit setting, another thread of work (Chen et al. 2009, Adamczyk 2011, Bansal et al. 2012) imposes an additional *per-vertex budget constraint* by which the algorithm is not allowed to query more than a specified number, b_v, of edges incident to vertex v. With this additional constraint, the benchmark with which the algorithm is compared switches from the omniscient optimum to the constrained optimum, that is, the performance of the best decision tree that obeys the per-vertex budget constraints and the query-commit restriction. In other words, the algorithm's disadvantage compared with the benchmark is only that it is constrained to run in polynomial time. Here, again, the best known approximation ratios are constant. A generalization of these results to packing problems has been studied by Gupta and Nagarajan (2013).

Similarly to our work, Blum et al. (2013) consider a stochastic matching setting without the query-commit constraint. They set the per-vertex budget to exactly two and ask which subset of edges is queried by the optimal collection of queries subject to this constraint. They prove structural results about the optimal solution, which allow them to show that finding the optimal subset of edges to query is NP-hard. In addition, they give a polynomial-time algorithm that finds an almost optimal solution on a class of random graphs (inspired by kidney exchange settings). Crucially, the benchmark of Blum et al. (2013) is also constrained to two queries per vertex.

There is a significant body of work in stochastic optimization more broadly, for instance, the papers of Dean et al. (2004) (stochastic knapsack), Gupta et al. (2012) (stochastic orienteering), and Asadpour et al. (2008) (stochastic submodular maximization).

The preliminary version of this paper (Blum et al. 2015) motivated a recent follow-up work by Assadi et al. (2016). In their work, Assadi et al. (2016) consider the stochastic matching (two-cycle packing) problem and show that preprocessing the graph before applying our algorithm achieves the same approximation guarantee using fewer queries per vertex. In particular, for both our adaptive and nonadaptive algorithms, the number of queries per vertex, even though independent of the number of vertices, is exponential in $1/\epsilon$. For the particular case of two-cycle matching, Assadi et al. (2016) show that performing a vertex sparsification step before applying our algorithm obtains a similar approximation guarantee, that is, $(1-\epsilon)$ for adaptive and $0.5(1-\epsilon)$ for the nonadaptive algorithms, using a number of queries that is polynomial in $1/\epsilon$.

2.2. Kidney Exchange

Early models of kidney exchange did not explicitly consider the setting in which an edge that is chosen to be matched only exists probabilistically. Recent research by Dickerson et al. (2019) and Anderson et al. (2015b) focuses on the kidney exchange application and restricts attention to a single crossmatch test per patient (the current practice) with a similar goal of maximizing the expected number of matched vertices in a realistic setting (for example, they allow three cycles and chains initiated by altruistic donors who enter the exchange without a paired patient). They develop integer programming techniques, which are empirically evaluated using real and synthetic data. As opposed to that line of work, which takes into account a single compatibility test per patient, our work considers the benefit that multiple tests per patient can bring to the quality of the matching. Manlove and O'Malley (2015) discuss the integer programming formulation used by the national exchange in the United Kingdom, which takes edge failures into account in an ad hoc way by, for example, preferring shorter cycles to longer ones. To our knowledge, our paper is the first to describe a general method for testing any number of edges *before* the final match run is performed and to provide experiments on real data showing the expected effect on fielded exchanges of such edge-querying policies.

Another form of stochasticity present in fielded kidney exchanges is the arrival and departure of donor–patient pairs over time (and the associated arrival and departure of their involved edges in the compatibility graph). Recent work has addressed this added form of dynamism from a theoretical (Ünver 2010, Akbarpour et al. 2014, Anderson et al. 2015a) and experimental (Awasthi and Sandholm 2009, Dickerson et al. 2012a, Dickerson and Sandholm 2015) point of view. Theoretical models have not addressed the case in which an edge in the current graph may not exist (as we do in this paper); the more recent experimental papers have incorporated this possibility but have not considered the problem of querying edges before recommending a final matching. We leave as future research the analysis of edge querying in stochastic matching in such a dynamic model.

3. The Model

For any graph $G = (V, E)$, let $M(E)$ denote its maximum (cardinality) matching. In the notation $M(E)$, we intentionally suppress the dependence on the vertex set V because we are only interested in the maximum matchings of different subsets of edges for a fixed vertex set. In addition, for two matchings M and M', we denote their *symmetric difference* by $M \Delta M' = (M \cup M') \setminus (M \cap M')$; it includes only paths and cycles consisting of alternating edges of M and M'.

In the stochastic setting, given a set of edges X, define X_p to be the random subset formed by including each edge of X independently with probability p. We assume for ease of exposition that $p_e = p$ for all edges $e \in E$. Our results hold when p is a lower bound, that is, $p_e \geq p$ for all $e \in E$. Furthermore, in Appendix EC.1, we show that we can extend our results to a more general setting in which the existence probabilities of edges incident to any particular vertex are correlated.

Given a graph $G = (V, E)$, define $\overline{M}(E)$ to be $\mathbb{E}[|M(E_p)|]$, where the expectation is taken over the random draw E_p. In addition, given the results of queries on some set of edges T, define $\overline{M}(E|T)$ to be $\mathbb{E}[|M(X_p \cup T')|]$, where $T' \subseteq T$ is the subset of edges of T that are known to exist based on the queries, and $X = E \setminus T$.

In the nonadaptive version of our problem, the goal is to design an algorithm that, given a graph $G = (V, E)$ with $|V| = n$, queries a subset X of edges in parallel such that $|X| = O(n)$ and maximizes the ratio $\overline{M}(X)/\overline{M}(E)$.

In contrast, an adaptive algorithm proceeds in rounds and, in each round, queries a subset of edges in parallel. Based on the results of the queries up to the current round, it can choose the subset of edges to test in the next round. Formally, an *R-round adaptive* stochastic matching algorithm selects, in each round r, a subset of edges $X_r \subseteq E$, where X_r can be a function of the results of the queries on $\bigcup_{i<r} X_i$. The objective is to maximize the ratio $\mathbb{E}[|M(\bigcup_{1\leq i\leq R} X_i)|]/\overline{M}(E)$, where the expectation in the numerator is taken over the outcome of the query results and the sets X_i chosen by the algorithm.

4. Understanding the Challenges

To gain some intuition for our goal of arbitrarily good approximations to the omniscient optimum and why it is challenging, let us consider a naïve algorithm and understand why it fails. This nonadaptive algorithm schedules $R = O(\log(n)/p)$ queries for each vertex as follows. First, order all vertices arbitrarily and start with an empty set of queries. For each vertex v, let $N_R(v)$ be the set of neighbors of v for whom at most R queries have been scheduled. Schedule $\min\{R, N_R(v)\}$ queries, each between v and an element of $N_R(v)$, where these elements are selected uniformly at random from $N_R(v)$.

The next example shows that this proposed algorithm only achieves $\frac{5}{6}$ fraction of the omniscient optimal solution as opposed to our goal of achieving arbitrarily good $(1-\epsilon)$ approximations to the omniscient optimal. Furthermore, in the following example, when each edge exists with probability $p > \frac{5}{6}$, this algorithm still only achieves a $\frac{5}{6}$ fraction of the omniscient optimal solution, which is worse than a trivial algorithm of just picking *one* maximum matching that guarantees a matching of size pn.

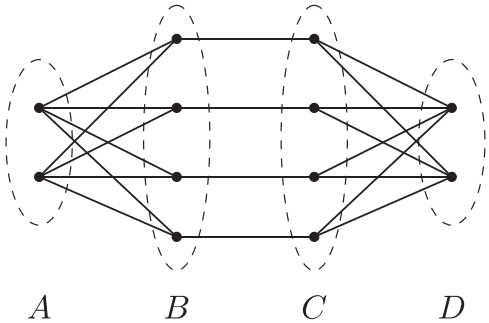
Example 1. Consider the graph $G = (V, E)$ whose vertices are partitioned into sets A, B, C, and D such that $|A| = |B| = \frac{n}{2}$ and $|C| = |D| = n$. Let E consist of two random bipartite graphs of degree $R = O(\log(n)/p)$ between A and B and similarly between C and D. And let B and C be connected with a complete bipartite graph. Let p be the existence probability of any edge.

With high probability, there is a perfect matching that matches A to B and C to D. For ease of exposition, consider the algorithm at the stage when half of the vertices have been processed. In expectation, half of the vertices in A, B, C, and D are processed by this point. Every vertex in B has more neighbors in C than in A. So, at this point, with high probability, all of the vertices of B already have R queries scheduled from half of the vertices in C. Therefore, after this point in the algorithm, no edges between A and B will be queried. So half of the vertices in A remain unmatched. Compared with the omniscient optimum—which is a perfect matching with high probability—the approximation ratio of this algorithm is at most $\frac{5}{6}$.

In the aforementioned example, $N_R(v)$ is restricted to vertices that have received at most R queries to bias the choice of queries toward vertices with fewer scheduled queries. At a high level, this is done to avoid scheduling queries for vertices that have already found existing and suitable matches. In the next example, we show that a naïve algorithm that uniformly queries $o(n)$ neighbors of each vertex—and, therefore, does not bias the queries toward vertices with fewer existing queries—suffers from even worse performance.

Example 2. Consider the graph $G = (V, E)$ whose vertices are partitioned into sets A, B, C, and D such that $|A| = |D| = t^\beta$ and $|B| = |C| = t$ for some $1 > \beta > 0$. Note that, in this graph, $n = \Theta(t)$. Let E consist of one perfect matching between the vertices of B and C and two complete bipartite graphs, one between A and B and another between C and D. See Figure 2 for an illustration. Let $p = 0.5$ be the existence probability of any edge.

Figure 2. Illustration of the Construction in Example 2 for $t = 4$ and $\beta = 1/2$



The omniscient optimal solution can use any edge, and in particular, it can use the edges between B and C. Because these edges form a matching of size t and $p = 0.5$, they alone provide a matching of expected size $t/2$. Hence, $\overline{M}(E) \geq t/2$.

Now, for any $\alpha < \beta$, consider the algorithm that queries t^α random neighbors for each vertex. For every vertex in B, the probability that its edge to C is chosen is at most $\frac{t^\alpha}{t^\beta+1}$ (similarly for the edges from C to B). Therefore, the expected number of edges chosen between B and C is at most $\frac{2t^{1+\alpha}}{t^\beta+1}$, and the expected number of existing edges between B and C, after the coin tosses, is at most $\frac{t^{1+\alpha}}{t^\beta+1}$. A and D each have t^β vertices, so they contribute at most $2t^\beta$ edges to any matching. Therefore, the expected size of the overall matching is no more than $t^{1+\alpha-\beta} + 2t^\beta$. Using $n = \Theta(t)$, we conclude that the approximation ratio of the naïve algorithm approaches zero as $n \to \infty$. For $\alpha = 0.5$ and $\beta = 0.75$, the approximation ratio of the naïve algorithm is $O(1/n^{0.25})$, at best.

5. Adaptive Algorithm: $(1-\epsilon)$-Approximation

In this section, we present our main result: an adaptive algorithm—formally given as Algorithm 1—that achieves a $(1-\epsilon)$ approximation to the omniscient optimum for arbitrarily small $\epsilon > 0$, using $O(1)$ queries per vertex and $O(1)$ rounds.

The algorithm is initialized with the empty matching M_0. At the end of each round r, our goal is to maintain a maximum matching M_r on the set of edges that are known to exist (based on queries made so far). To this end, at round r, we compute the maximum matching O_r on the set of edges that are known to exist *and* the ones that have not been queried yet (step 2a). We consider augmenting paths in $O_r \Delta M_{r-1}$ and query all the edges in them (steps 2b and 2c). Based on the results of these queries (Q_r), we update the maximum matching (M_r). Finally, we return the maximum matching M_R computed after $R = \frac{\log(2/\epsilon)}{p^{2/\epsilon}}$ rounds. (Let us assume that R is an integer for ease of exposition.)

Algorithm 1 (Adaptive Algorithm for Stochastic Matching: $(1-\epsilon)$-Approximation)

Input: A graph $G = (V, E)$.

Parameter: $R = \frac{\log(2/\epsilon)}{p^{2/\epsilon}}$.

1. Initialize M_0 to the empty matching and $W_1 \leftarrow \emptyset$.
2. For $r = 1, \ldots, R$, do
   a. Compute a maximum matching, O_r, in $(V, E \setminus W_r)$.
   b. Set Q_r to the collection of all augmenting paths of M_{r-1} in $O_r \Delta M_{r-1}$.
   c. Query the edges in Q_r. Let Q'_r and Q''_r represent the set of existing and nonexisting edges.
   d. $W_{r+1} \leftarrow W_r \cup Q''_r$.
   e. Set M_r to the maximum matching in $\left(V, \cup_{j=1}^{r} Q'_j\right)$.
3. Output M_R.

It is easy to see that *this algorithm queries at most* $\frac{\log(2/\epsilon)}{p^{2/\epsilon}}$ *edges per vertex*. In a given round r, the algorithm queries edges that are in augmenting paths of $O_r \Delta M_{r-1}$. Because there is at most one augmenting path using any particular vertex, the algorithm queries at most one edge per vertex in each round. Furthermore, the algorithm executes $\frac{\log(2/\epsilon)}{p^{2/\epsilon}}$ rounds. Therefore, the number of queries issued by the algorithm per vertex is as claimed.

The rest of the section is devoted to proving that the matching returned by this algorithm after R rounds has cardinality that is, in expectation, at least a $(1-\epsilon)$ fraction of $\overline{M}(E)$.

Theorem 1. *For any graph $G = (V, E)$ and any $\epsilon > 0$, Algorithm 1 returns a matching whose expected cardinality is at least $(1-\epsilon)\overline{M}(E)$ in $R = \frac{\log(2/\epsilon)}{p^{(2/\epsilon)}}$ rounds.*

As mentioned in Section 1, one of the insights behind this result is the existence of many disjoint augmenting paths of *bounded length* that can be used to augment a matching that is far from the omniscient optimum, that is, a lower bound on the number of elements in Q_r of a given length L. This observation is formalized in the following lemma. (We emphasize that the lemma pertains to the nonstochastic setting.)

Lemma 1. *Consider a graph $G = (V, E)$ with two matchings M_1 and M_2. Suppose $|M_2| > |M_1|$. Then, in $M_1 \Delta M_2$, for any odd length $L \geq 1$, there exist at least $|M_2| - (1 + \frac{2}{L+1})|M_1|$ augmenting paths of length at most L, which augment the cardinality of M_1.*

Proof. Let x_l be the number of augmenting paths of length l (for any odd $l \geq 1$) found in $M_1 \Delta M_2$ that augment the cardinality of M_1. Each augmenting path increases the size of M_1 by one, so the total number of augmenting paths $\sum_{l \geq 1} x_l$ is at least $|M_2| - |M_1|$. Moreover, each augmenting path of length l has $\frac{l-1}{2}$ edges in M_1. Hence, $\sum_{l \geq 1} \frac{l-1}{2} x_l \leq |M_1|$. In particular, this implies that $\frac{L+1}{2} \sum_{l \geq L+2} x_l \leq |M_1|$. We conclude that

$$\sum_{l=1}^{L} x_l = \sum_{l \geq 1} x_l - \sum_{l \geq L+2} x_l \geq (|M_2| - |M_1|) - \frac{2}{L+1}|M_1|$$
$$= |M_2| - \left(1 + \frac{2}{L+1}\right)|M_1|. \quad \square$$

The rest of the theorem's proof requires some additional notation. At the beginning of any given round r, the algorithm already knows about the existence (or nonexistence) of the edges in $\cup_{i=1}^{r-1} Q_i$. We use Z_r to denote the expected size of the maximum matching in graph $G = (V, E)$ given the results of the queries $\cup_{i=1}^{r-1} Q_i$. More formally, $Z_r = \overline{M}(E | \cup_{i=1}^{r-1} Q_i)$. Note that $Z_1 = \overline{M}(E)$.

For a given r, we use the notation $\mathbb{E}_{Q_r}[X]$ to denote the expected value of X, where the expectation is taken *only* over the outcome of query Q_r and fixing the outcomes on the results of queries $\bigcup_{i=1}^{r-1} Q_i$. Moreover, for a given r, we use $\mathbb{E}_{Q_r,\dots,Q_R}[X]$ to denote the expected value of X with the expectation taken over the outcomes of queries $\bigcup_{i=r}^{R} Q_i$ and fixing an outcome on the results of queries $\bigcup_{i=1}^{r-1} Q_i$.

In Lemma 2, for any round r and for *any* outcome of the queries $\bigcup_{i=1}^{r-1} Q_i$, we lower bound the *expected increase in the size of* M_r over the size of M_{r-1} with the expectation being taken only over the outcome of edges in Q_r. This lower bound is a function of Z_r.

Lemma 2. *For any $r \in [R]$, odd L, and $Q_1, \cdots, Q_{r-1}$, it holds that $\mathbb{E}_{Q_r}[|M_r|] \geq (1-\gamma)|M_{r-1}| + \alpha Z_r$, where $\gamma = p^{(L+1)/2}\left(1+\frac{2}{L+1}\right)$ and $\alpha = p^{(L+1)/2}$.*

Proof. By Lemma 1, there exist at least $|O_r| - (1+\frac{2}{L+1})|M_{r-1}|$ augmenting paths in $O_r \Delta M_{r-1}$ that augment M_{r-1} and are of length at most L. The O_r part of every augmenting path of length at most L exists independently with probability at least $p^{(L+1)/2}$. Therefore, the expected increase in the size of the matching is

$$\begin{aligned}\mathbb{E}_{Q_r}[|M_r|] - |M_{r-1}| &\geq p^{\frac{L+1}{2}}\left(|O_r| - \left(1+\frac{2}{L+1}\right)|M_{r-1}|\right)\\ &= \alpha|O_r| - \gamma|M_{r-1}| \geq \alpha Z_r - \gamma|M_{r-1}|,\end{aligned}$$

where the last inequality holds by the fact that Z_r, which is the expected size of the optimal matching with expectation taken over nonqueried edges, cannot be larger than O_r, which is the maximum matching, assuming that every nonqueried edge exists. □

We are now ready to prove the theorem.

Proof of Theorem 1. Let $L = \frac{4}{\epsilon} - 1$; it is assumed to be an odd integer for ease of exposition. Otherwise, there exists $\epsilon/2 \leq \epsilon' \leq \epsilon$ such that $\frac{4}{\epsilon'} - 1$ is an odd integer. We use a similar simplification in the proofs of other results. By Lemma 2, we know that, for every $r \in [R]$, $\mathbb{E}_{Q_r}[||M_r|| \geq (1-\gamma)||M_{r-1}|| + \alpha Z_r$, where $\gamma = p^{(L+1)/2}(1+\frac{2}{L+1})$, and $\alpha = p^{(L+1)/2}$. We use this inequality repeatedly to derive our result. We also require the equality

$$\mathbb{E}_{Q_{r-1}}[Z_r] = \mathbb{E}_{Q_{r-1}}\left[\overline{M}\left(E|\bigcup_{i=1}^{r-1} Q_i\right)\right] = \overline{M}\left(E|\bigcup_{i=1}^{r-2} Q_i\right) = Z_{r-1}. \tag{1}$$

First, applying Lemma 2 at round R, we have that $\mathbb{E}_{Q_R}[|M_R|] \geq (1-\gamma)|M_{R-1}| + \alpha Z_R$. This inequality is true for any fixed outcomes of $Q_1, \dots, Q_{R-1}$. In particular, we can take the expectation over Q_{R-1} and obtain

$$\mathbb{E}_{Q_{R-1},Q_R}[|M_R|] \geq (1-\gamma)\mathbb{E}_{Q_{R-1}}[|M_{R-1}|] + \alpha\mathbb{E}_{Q_{R-1}}[Z_R].$$

By Equation (1), we know that $\mathbb{E}_{Q_{R-1}}[Z_R] = Z_{R-1}$. Furthermore, we can apply Lemma 2 to $\mathbb{E}_{Q_{R-1}}[|M_{R-1}|]$ to get the following inequality:

$$\begin{aligned}\mathbb{E}_{Q_{R-1},Q_R}[|M_R|] &\geq (1-\gamma)\mathbb{E}_{Q_{R-1}}[|M_{R-1}|] + \alpha\mathbb{E}_{Q_{R-1}}[Z_R]\\ &\geq (1-\gamma)\left((1-\gamma)|M_{R-2}| + \alpha Z_{R-1}\right) + \alpha Z_{R-1}\\ &= (1-\gamma)^2|M_{R-2}| + \alpha(1+(1-\gamma))Z_{R-1}.\end{aligned}$$

We repeat these steps by sequentially taking expectations over Q_{R-2} through Q_1 and, at each step, applying Equation (1) and Lemma 2. This gives us

$$\begin{aligned}\mathbb{E}_{Q_1,\dots,Q_R}[|M_R|] &\geq (1-\gamma)^R|M_0| + \alpha\Big(1+(1-\gamma)\\ &\quad +\cdots+(1-\gamma)^{R-1}\Big)Z_1\\ &= \alpha\frac{1-(1-\gamma)^R}{\gamma}Z_1,\end{aligned}$$

where the second transition follows from the initialization of M_0 as an empty matching. Because $L = \frac{4}{\epsilon} - 1$ and $R = \frac{\log(2/\epsilon)}{p^{2/\epsilon}}$, we have

$$\begin{aligned}\frac{\alpha}{\gamma}\left(1-(1-\gamma)^R\right) &= \left(1-\frac{2}{L+1}\right)\left(1-(1-\gamma)^R\right)\\ &\geq 1-\frac{2}{L+1}-e^{-\gamma R} \geq 1-\frac{\epsilon}{2}-\frac{\epsilon}{2}\\ &= 1-\epsilon,\end{aligned} \tag{2}$$

where the second transition is true because $e^{-x} \geq 1-x$ for all $x \in \mathbb{R}$. We conclude that $\mathbb{E}_{Q_1,\dots,Q_R}[|M_R|] \geq (1-\epsilon)Z_1$. Because $Z_1 = \overline{M}(E)$, it follows that the expected size of the algorithm's output is at least $(1-\epsilon)\overline{M}(E)$. □

In Appendix EC.1, we extend our results to the setting in which edges have correlated existence probabilities; an edge's probability is determined by parameters associated with its two vertices. This generalization gives a better model for kidney exchange as some patients are *highly sensitized* and, therefore, harder to match in general; this means that all edges incident to such vertices are less likely to exist. We consider two settings: the first in which an adversary chooses the vertex parameters and the second in which these parameters are drawn from a distribution. Our approach involves excluding from our analysis edges whose existence probability is too low. We do so by showing that (under specific conditions) excluding any augmenting path that includes such edges still leaves us with a large number of constant-size augmenting paths.

6. Nonadaptive Algorithm: 0.5-Approximation

The adaptive algorithm, Algorithm 1, augments the current matching by computing a maximum matching

on queried edges that are known to exist and edges that have not been queried. One way to extend this idea to the nonadaptive setting is the following: we can simply choose several edge-disjoint matchings and hope that they help in augmenting each other. In this section, we ask: how close can this nonadaptive interpretation of our adaptive approach take us to the omniscient optimum?

In more detail, our nonadaptive algorithm—formally given as Algorithm 2—iterates $R = \frac{\log(2/\epsilon)}{p^{2/\epsilon}}$ times. In each iteration, it picks a maximum matching and removes it. The set of edges queried by the algorithm is the union of the edges chosen in some iteration. We show that, for any arbitrarily small $\epsilon > 0$, the algorithm finds a $0.5(1-\epsilon)$-approximate solution. Because we allow an arbitrarily small (although constant) probability p for stochastic matching, achieving a 0.5-approximation independently of the value of p while querying only a linear number of edges is nontrivial. For example, a naïve algorithm that only queries one maximum matching clearly does not guarantee a 0.5-approximation; it would guarantee only a p-approximation. In addition, the example given in Section 3 shows that choosing edges at random performs poorly.

Algorithm 2 (Nonadaptive Algorithm for Stochastic Matching: 0.5-Approximation)
Input: A graph $G(V, E)$.
Parameter: $R = \frac{\log(2/\epsilon)}{p^{2/\epsilon}}$.

1. Initialize $W_0 \leftarrow \emptyset$.
2. For $r = 1, \ldots, R$, do
   a. Compute a maximum matching, O_r, in $(V, E \setminus W_{r-1})$.
   b. $W_r \leftarrow W_{r-1} \cup O_r$.
3. Query all the edges in W_R and output the maximum matching among the edges that are found to exist in W_R.

The number of edges incident to any particular vertex that are queried by the algorithm is at most $\frac{\log(2/\epsilon)}{p^{2/\epsilon}}$ because the vertex can be matched with at most one neighbor in each round. The next theorem establishes the approximation guarantee of Algorithm 2.

Theorem 2. *Given a graph $G = (V, E)$ and any $\epsilon > 0$, the expected size $\overline{M}(W_R)$ of the matching produced by Algorithm 2 is at least a $0.5(1-\epsilon)$ fraction of $\overline{M}(E)$.*

Similar to the adaptive procedure of Section 5, the proof of Theorem 2 relies on analyzing how much matching O_r increases the size of the expected matching, $\overline{M}(W_{r-1})$, at every round. As opposed to the adaptive procedure, here we do not query the edges in W_{r-1}. Hence, we need to reason about the *expected size* of the matching up to round r, $\overline{M}(W_{r-1})$, and the expected size of the matching in the remaining graph, $\overline{M}(E \setminus W_{r-1})$. The following lemma can be used to bound $\overline{M}(E \setminus W_{r-1})$ in terms of $\overline{M}(W_{r-1})$ and $\overline{M}(E)$.

Lemma 3. *Let E_1 be an arbitrary subset of edges of E and let $E_2 = E \setminus E_1$. Then $\overline{M}(E) \le \overline{M}(E_1) + \overline{M}(E_2)$.*

Proof. Let E' be an arbitrary subset of edges of E and let $E'_1 = E_1 \cap E'$ and $E'_2 = E_2 \cap E'$. We claim that $|M(E')| \le |M(E'_1)| + |M(E'_2)|$. This is because, if T is the set of edges in a maximum matching in graph (V, E'), then clearly $T \cap E'_1$ and $T \cap E'_2$ are valid matchings in E'_1 and E'_2, respectively, and thereby it follows that $|M(E'_1)| \ge |T \cap E'_1|$ and $|M(E'_2)| \ge |T \cap E'_2|$, and hence, $|M(E')| \le |M(E'_1)| + |M(E'_2)|$. Expectation is a convex combination of the values of the outcomes. For every subset E' of edges in E, multiplying the previous inequality by the probability that the outcome of the coin tosses on the edges of E is E' and then summing the various inequalities, we get $\overline{M}(E) \le \overline{M}(E_1) + \overline{M}(E_2)$. □

To lower bound $\overline{M}(W_R)$, we first show that, for any round r, either our current collection of edges has an expected matching size $\overline{M}(W_{r-1})$ that compares well with $\overline{M}(E)$, or in round r, we have a significant increase in $\overline{M}(W_r)$ over $\overline{M}(W_{r-1})$.

Lemma 4. *At any iteration $r \in [R]$ of Algorithm 2 and odd L, if $\overline{M}(W_{r-1}) \le \overline{M}(E)/2$, then*

$$\overline{M}(W_r) \ge \frac{\alpha}{2}\overline{M}(E) + (1-\gamma)\overline{M}(W_{r-1}),$$

where $\gamma = p^{(L+1)/2}(1 + \frac{2}{L+1})$ and $\alpha = p^{(L+1)/2}$.

Proof. Assume that $\overline{M}(W_{r-1}) \le \overline{M}(E)/2$. By Lemma 3, we know that $\overline{M}(E \setminus W_{r-1}) \ge \overline{M}(E) - \overline{M}(W_{r-1})$. Recall that O_r is the maximum matching left in $E \setminus W_{r-1}$; therefore, $|O_r| = |M(E \setminus W_{r-1})| \ge \overline{M}(E \setminus W_{r-1}) \ge \overline{M}(E) - \overline{M}(W_{r-1}) \ge \overline{M}(E)/2$.

In a thought experiment, say at the beginning of round r, we query the set W_{r-1} and let W'_{r-1} be the set of edges that are found to exist. By Lemma 1, there are at least $|O_r| - (1 + \frac{2}{L+1})|M(W'_{r-1})|$ augmenting paths of length at most L in $O_r \Delta M(W'_{r-1})$ that augment $M(W'_{r-1})$. Each of these paths succeeds with probability at least $p^{(L+1)/2}$. We have

$$\begin{aligned}
&\overline{M}\big(O_r \cup W'_{r-1} | W'_{r-1}\big) - |M\big(W'_{r-1}\big)| \\
&\quad \ge p^{(L+1)/2}\left(|O_r| - (1 + \frac{2}{L+1})|M\big(W'_{r-1}\big)|\right) \qquad (3)\\
&\quad \ge p^{(L+1)/2}\left(\frac{1}{2}\overline{M}(E) - \left(1 + \frac{2}{L+1}\right)|M\big(W'_{r-1}\big)|\right), \qquad (4)
\end{aligned}$$

where the expectation on the left-hand side is taken only over the outcome of the edges in O_r. Therefore, we have $\overline{M}\big(O_r \cup W'_{r-1} | W'_{r-1}\big) \ge \frac{\alpha}{2}\overline{M}(E) + (1-\gamma)|M(W'_{r-1})|$, where $\alpha = p^{(L+1)/2}$ and $\gamma = p^{(L+1)/2}(1 + \frac{2}{L+1})$. Taking

expectation over the coin tosses on W_{r-1} that create outcome W'_{r-1}, we have our result, that is,

$$\overline{M}(W_r) \geq \mathbb{E}_{W_{r-1}}\left[\overline{M}(O_r \cup W'_{r-1} | W'_{r-1})\right] \geq \overline{M}(O_r \cup W_{r-1})$$
$$\geq \frac{\alpha}{2}\overline{M}(E) + (1-\gamma)\overline{M}(W_{r-1}). \quad \square$$

We are now ready to prove Theorem 2.

Proof of Theorem 2. For ease of exposition, assume $L = \frac{4}{\epsilon} - 1$ is an odd integer. Then, $\overline{M}(W_R) \geq \overline{M}(E)/2$, in which case, we are done, or otherwise, by repeatedly applying Lemma 4 for R steps, we have

$$\overline{M}(W_R) \geq \frac{\alpha}{2}\left(1 + (1-\gamma) + (1-\gamma)^2 + \cdots + (1-\gamma)^{R-1}\right)$$
$$\cdot \overline{M}(E) \geq \frac{\alpha}{2}\frac{(1-(1-\gamma)^R)}{\gamma}\overline{M}(E).$$

Now, $\frac{\alpha}{\gamma}(1-(1-\gamma)^R) \geq 1 - \frac{2}{L+1} - e^{-\gamma R} \geq 1 - \epsilon$ for $R = \frac{\log(2/\epsilon)}{p^2/\epsilon}$. Hence, we have our $0.5(1-\epsilon)$ approximation. $\square$

6.1. Upper Bound on the Performance of the Nonadaptive Algorithm

As we explain in more detail in Section 9.1, we do not know whether, in general, nonadaptive algorithms can achieve a $(1-\epsilon)$-approximation with $O_\epsilon(1)$ queries per vertex. However, if there is such an algorithm, it is not Algorithm 2! Indeed, the next theorem shows that the algorithm cannot give an approximation ratio better than 11/12 to the omniscient optimum. This fact holds even when $R = \Theta(\log n)$.

Theorem 3. *Let $p = 0.5$. For any $\epsilon > 0$, there exists n and a graph (V, E) with $|V| \geq n$ such that Algorithm 2 with $R = O(\log n)$ returns a matching with expected size of at most $\frac{11}{12}\overline{M}(E) + \epsilon$.*

Before proving Theorem 3, let us first show that the expected size of a matching, that is, $\overline{M}(E)$, is large in a complete bipartite graph.

Lemma 5. *Let $G = (U \cup V, U \times V)$ be a complete bipartite graph between U and V with $|U| = |V| = n$. For any constant probability p, $\overline{M}(E) \geq n - o(n)$.*

Proof. Denote by E_p the random set of edges formed by including each edge in $U \times V$ independently with probability p. We show that, with probability at least $1 - \frac{1}{n^8}$, over the draw E_p, the maximum matching in the graph $(U \cup V, E_p)$ is at least $n - c\log(n)$, where $c = 10/\log(\frac{1}{1-p})$, and this completes our claim.

To show this, we prove that, with probability at least $1 - \frac{1}{n^8}$, over the draw E_p, all subsets $S \subseteq U$ of size at most $n - c\log(n)$ have a neighborhood of size at least $|S|$. By Hall's theorem, our claim follows.

Consider any set $S \subseteq U$ of size at most $n - c\log(n)$. We call set S "bad" if there exists some set $T \subseteq V$ of size $(|S| - 1)$ such that S does not have edges to $V \setminus T$. Fix any set $T \subseteq V$ of size $|S| - 1$. Over draws of E_p, the probability that S has no outgoing edges to $V \setminus T$ is at most $(1-p)^{|S||V\setminus T|} = (1-p)^{|S|(n-|S|+1)}$. Hence, by union bound, the probability that S is bad is at most $\binom{n}{|S|-1}(1-p)^{|S|(n-|S|+1)}$.

Again, by union bound, the probability that some set $S \subseteq U$ of size at most $n - c\log(n)$ is bad is at most $\sum_{1\leq|S|\leq n-c\log(n)}\binom{n}{|S|}\binom{n}{|S|-1}(1-p)^{|S|(n-|S|+1)}$, and this, in turn, is at most

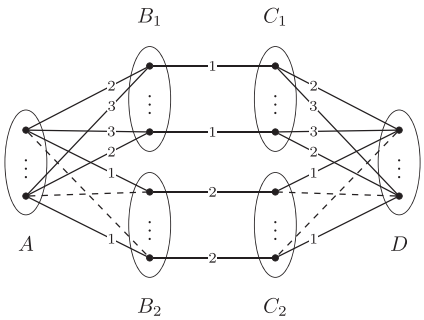
$$\sum_{1\leq|S|\leq n-c\log(n)} n^{|S|}n^{|S|}(1-p)^{|S|(n-|S|+1)}$$
$$\leq \sum_{1\leq|S|\leq n-c\log(n)} e^{|S|\cdot(2\log(n)+(n+1)\log(1-p)-|S|\log(1-p))}.$$

Note that the exponent in the summation achieves its maximum for $|S| = 1$. For $c = 10/\log(\frac{1}{1-p})$, we have that the given sum is at most $\exp(-\frac{n}{2}\log(\frac{1}{1-p}))$, and hence, with high probability, no set $S \subseteq U$ of size at most $n - c\log(n)$ is bad. $\square$

Proof of Theorem 3. Let (V, E) be a graph, illustrated in Figure 3, whose vertices are partitioned into sets A, B, C, and D such that $|A| = |D| = \frac{t}{2}$, $|B| = |C| = t$. The edge set E consists of one perfect matching between vertices of B and C and two complete bipartite graphs, one between A and B and another between C and D. Let $p = 0.5$ be the existence probability of any edge.

We first examine the value of the omniscient optimal, $\overline{M}(E)$. Because $p = 0.5$, in expectation, half of the edges in the perfect matching between B and C exist, and therefore, half of the vertices of B and C get matched. As we showed in Lemma 5, with high probability, the complete bipartite graph between the remaining half of B and A has a matching of size at least $t/2 - o(t)$. And similarly, with high probability, the complete bipartite graph between remaining half of C and D has a

Figure 3. Illustration of the Graph Used in the Proof of Theorem 3



Notes. Edges marked 1, 2, and 3 represent the matchings at rounds 1, 2, and 3, respectively. The dashed edges are never picked by the algorithm.

matching of size at least $t/2 - o(t)$. Therefore, $\overline{M}(E)$ is at least $\frac{3}{2}t - o(t)$.

Next, we look at Algorithm 2. For ease of exposition, let B_1 and B_2 denote the top and bottom halves of the vertices in B. Similarly, define C_1 and C_2. Because Algorithm 2 picks maximum matchings arbitrarily, we show that there exists a way of picking maximum matchings such that the expected matching size of the union of the edges picked in the matching is at most $\frac{11}{8}t$ $(= \frac{11}{12}\frac{3}{2}t)$.

Consider the following choice of maximum matching picked by the algorithm: In the first round, the algorithm picks the perfect matching between B_1 and C_1 and a perfect matching between A and B_2 and a perfect matching between C_2 and D. In the second round, the algorithm picks the perfect matching between B_2 and C_2 and a perfect matching each between A and B_1 and between C_1 and D. After these two rounds, we can see that there are no more edges left between B and C. For the subsequent $R-2$ rounds, in each round, the algorithm picks a perfect matching between A and B_1 and a perfect matching between C_1 and D. It is easy to verify that in every round, the algorithm has picked a maximum matching from the remnant graph.

We analyze the expected size of matching output by the algorithm. For each of the vertices in B_2 and C_2, the algorithm has picked only two incident edges. For any vertex in B_2 and C_2 with probability at least $(1-p)^2 = \frac{1}{4}$, none of these two incident edges exist. Hence, the expected number of vertices that are *unmatched* in B_2 and C_2 is at least $\frac{1}{4}(\frac{t}{2} + \frac{t}{2}) = \frac{t}{4}$. Hence, the total number of edges included in the matching is at most $\frac{1}{2}(3t - t/4) = \frac{11}{8}t$. This completes our claim. □

Despite this somewhat negative result, in Section 8, we show experimentally on realistic kidney exchange compatibility graphs that Algorithm 2 performs very well for even very small values of R across a wide range of values of p.

7. Generalization to Stochastic k-Cycle Packing

So far, we have focused on stochastic matching, with which the goal is equivalent to finding the largest two-cycle packing. In this section, we generalize our approach to the case of k-cycle packing for any $k \geq 2$.

Formally, for a directed graph $G = (V, E)$, the corresponding k-cycle packing instance (V, A) consists of the set of vertices V and the collection $A \subseteq V^{\leq k}$ of vertices that form a directed cycle of length at most k in G, where $V^{\leq k} = \bigcup_{i=1}^{k} V^i$. Given graph G and its corresponding k-cycle packing instance (V, A), a feasible solution to the k-cycle packing instance is a collection $B \subseteq A$ such that the cycles in B are vertex-disjoint. Let $V(A) \subseteq V$ denote the largest set of vertices that can be covered by a feasible k-cycle packing $B \subseteq A$, that is, vertices in $\bigcup_{c \in B} c$. Moreover, let $K(A)$ denote the feasible k-cycle packing B with largest $|B|$.

In the stochastic variant of k-cycle packing, given a graph $G = (V, E)$, we represent by $E_p \sim E$ a random subset of edges in which each edge in E is included in E_p with probability p independently. We represent by $(V, A(E_p))$ the k-cycle packing instance that corresponds to the graph (V, E_p). Note that, for any E_p, $A(E_p) \subseteq A$ is the set of those cycles in A whose edges appear in E_p. We denote by $\overline{V}(A) = \mathbb{E}_{E_p \sim E}\left[\left|V(A(E_p))\right|\right]$ and $\overline{K}(A) = \mathbb{E}_{E_p \sim E}\left[\left|K(A(E_p))\right|\right]$, respectively, the expected maximum number of vertices covered in a k-cycle packing and the expected maximum cardinality of a k-cycle packing.

Note that our goal in kidney exchange is to match the largest number of donor–patient pairs; therefore, our omniscient optimum benchmark is $\overline{V}(A)$. However, a k-cycle packing B such that $|B| \geq \alpha\overline{K}(A)$ covers a number of vertices that is at least $\frac{2}{k}\alpha\overline{V}(A)$ because every cycle in B covers at least two vertices and the cycles in A cover at most k vertices each. Therefore, for the majority of this section, we focus on finding a k-cycle packing whose expected size is a good approximation of $\overline{K}(A)$, and as a result, the number of vertices covered by it is a good approximation of $\overline{V}(A)$. We present a polynomial-time adaptive algorithm, Algorithm 4, that obtains a $(1-\epsilon)\frac{2}{k}$-approximation of $\overline{K}(A)$ and a $(1-\epsilon)\frac{4}{k^2}$-approximation of $\overline{V}(A)$.

Theorem 4. *There exists an adaptive polynomial-time algorithm that, given a graph $G = (V, E)$, its corresponding k-cycle packing instance (V, A), and $\epsilon > 0$, uses $R = O_{\epsilon,k}(1)$ rounds and $O_{\epsilon,k}(n)$ edge queries overall and returns a cycle-packing B_R such that $|B_R| \geq (1-\epsilon)\frac{2}{k}\overline{K}(A)$. Moreover, $\sum_{c \in B_R} |c| \geq (1-\epsilon)\frac{4}{k^2}\overline{V}(A)$.*

Importantly, the statement of Theorem 1 for adaptive stochastic matching is a special case of the statement of Theorem 4 for $k = 2$. By contrast, we leave the case of nonadaptive algorithms for k-cycle packing for general $k \geq 2$ as an open problem—despite having presented Theorem 2 for the special case of $k = 2$—and describe some of the challenges one may face in obtaining such a general result for $k > 2$ in Section 9.1.

7.1. Augmenting Structures for k-Cycle Packing

Finding an optimal solution to the k-cycle packing problem is NP-hard (Abraham et al. 2007). On the other hand, multiple approximation algorithms are known for k-cycle packing and its generalization to k-set packing, in which A includes arbitrary subsets of $\leq k$ elements of V. One such algorithm is a local search algorithm of Hurkens and Schrijver (1989) that uses a notion of *augmenting structures*. Given a k-cycle packing instance (V, A) and a feasible packing (one with disjoint cycles) $B \subseteq A$, (C, D) is said to be an augmenting structure for B if removing D and

adding C to B increases its cardinality and maintains the feasibility of the packing; that is, if $(B \cup C) \setminus D$ is a collection of vertex-disjoint k-cycles and $|(B \cup C) \setminus D| > |B|$, where $C \subseteq A$ and $D \subseteq B$.

Hurkens and Schrijver (1989) show that, for any η, there is a polynomial-time (assuming η and k are constants) approximation algorithm that repeatedly augments a feasible solution using augmenting structures of size $s_{\eta,k}$, a constant that depends on k and η, and obtains a packing of cardinality $\geq (\frac{2}{k} - \eta)K(A)$. Hurkens and Schrijver (1989) also show that an approximation ratio better than $2/k$ cannot be achieved with local search of structures of constant size. The following theorem summarizes the results of Hurkens and Schrijver (1989).

Lemma 6 (Hurkens and Schrijver (1989)). *Given a k-cycle packing instance (V, A) and a feasible packing B such that $|B| < (\frac{2}{k} - \eta)|K(A)|$, there exists an augmenting structure (C, D) for B such that $|C| \leq |s_{\eta,k}|$ and $|D| \leq s_{\eta,k}$ for some constant $s_{\eta,k}$ that depends only on η and k.*

Similarly to the case of two-cycle matchings, we require *many* augmenting structures because some may fail to exist when edges appear at random. In the following lemma, we show how to find a large number of small augmenting structures.

Lemma 7. *Given a k-cycle packing instance (V, A) and a feasible packing B such that $|B| < (\frac{2}{k} - \eta)|K(A)|$, there are $T = \frac{1}{ks_{\eta,k}} \left(|K(A)| - \frac{|B|}{\frac{2}{k} - \eta}\right)$ augmenting structures $(C_1, D_1), \ldots, (C_T, D_T)$ such that $|C_t| \leq s_{\eta,k}$ and $|D_t| \leq s_{\eta,k}$ for all $t \in [T]$, and the set of cycles appearing in C_ts are vertex-disjoint, that is, for all $t, t' \in [T]$ such that $t \neq t'$ for any $c \in C_t$ and $c' \in C_{t'}$, we have $c \cap c' = \emptyset$. Moreover, this collection of augmenting structures can be found in polynomial time, assuming k and η to be constants.*

Algorithm 3 (Finding Constant-Size Disjoint Augmenting Structures for k-Cycles)
Input: k-cycle packing instance (V, A) and a collection $B \subseteq A$ of disjoint sets.
Output: Collection Q of vertex-disjoint augmenting structures as described in Lemma 7.
Parameter: $s_{\eta,k}$ (the desired maximum size of the augmenting structures).
1. Initialize $A_1 \leftarrow A$ and $Q \leftarrow \phi$ (empty set).
2. For $t = 1, \cdots, |A|$
   a. Find an augmenting structure (C_t, D_t) of size $s_{\eta,k}$ for B on the k-cycle packing instance (V, A_t).
   b. Add (C_t, D_t) to Q. (If C_t is an empty set, break out of the loop.)
   c. $A_{t+1} \leftarrow A_t \setminus \{c | \exists c' \in C_t, \text{such that } c \cap c' \neq \emptyset\}$.
3. Output Q.

Proof. We prove this lemma using Algorithm 3. The algorithm starts with an empty set of augmenting structures Q. In step 2b of this algorithm, an augmenting structure for B gets added to Q. Furthermore, in step 2c, all cycles that share any vertex with a cycle that is already in Q are removed. Therefore, by design, the collection Q of augmenting structures returned by the algorithm satisfies the property that, for any two augmenting structures $C_t, C_{t'}$ and any two cycles $c \in C_t$ and $c' \in C_{t'}$, $c \cap c' = \emptyset$. It remains to show that $|Q| \geq \frac{1}{ks_{\eta,k}}(|K(A)| - \frac{|B|}{\frac{2}{k} - \eta})$. That is, in the first $T = \frac{1}{ks_{\eta,k}} \left(|K(A)| - \frac{|B|}{\frac{2}{k} - \eta}\right)$ iterations of step 2a, we are able to find a nonempty augmenting structure for B. Using Lemma 6, it is sufficient to show that $K(A_{t+1}) \geq |B|/(\frac{2}{k} - \eta)$ for all $t \leq T$.

Note that, for all t, $|C_t| \leq s_{\eta,k}$ and for each $c \in C_t$, $|c| \leq k$. Therefore, by time $t + 1$, there are at most $t \cdot k \cdot s_{\eta,k}$ vertices of V that are covered by cycles that appear in $C_1, \ldots, C_t$. At most $t \cdot k \cdot s_{\eta,k}$ cycles in the largest cardinality packing of A may have one or more of these $t \cdot k \cdot s_{\eta,k}$ vertices. Note that removing these cycles from the largest cardinality packing $K(A)$ yields a packing for A_{t+1}, so we have that

$$K(A_{t+1}) \geq K(A) - t \cdot k \cdot s_{\eta,k} \geq \frac{|B|}{\frac{2}{k} - \eta},$$

for all $t \leq T$. Using Lemma 6 completes the proof. □

7.2. Adaptive Algorithm for k-Cycle Packing

We use the following polynomial-time algorithm for stochastic k-cycle packing to prove Theorem 4. In each round r, the algorithm maintains a feasible k-cycle packing B_r based on the k-cycles that have been queried so far. It then computes a collection Q_r of vertex-disjoint, small augmenting structures with respect to the current solution B_r (as in Lemma 7), where the augmenting structures are composed of cycles that may have unqueried edges. It then queries all edges that appear in some cycle in these augmenting structures and uses those that are found to exist to augment the current solution and removes all cycles with a failed edge from consideration for the future rounds. The augmented solution is fed into the next round, and the process is repeated.

Algorithm 4 (Adaptive Algorithm for Stochastic k-Cycle Packing)
Input: Graph $G = (V, E)$, $k \geq 2$, the corresponding k-cycle packing instance (V, A), and $\epsilon > 0$.
Parameters: $\eta = \frac{\epsilon}{k}$ and $R = \frac{(\frac{2}{k} - \eta)k \cdot s_{\eta,k}}{p^{k \cdot s_{\eta,k}}} \log(\frac{2}{\epsilon})$ (for a $(1 - \epsilon)(\frac{2}{k})$-approximation to $\overline{K}(A)$).
1. Initialize $r \leftarrow 1$, $B_1 \leftarrow \emptyset$, and $A_1 \leftarrow A$.
2. For $r = 1, \ldots, R$, do
   a. Let Q_r be the set of augmenting structures given by Algorithm 3 on the input of the k-cycle packing instance of (V, A_r), the collection B_r, and the parameter $s_{\eta,k}$.

b. For each augmenting structure $(C, D) \in Q_r$
   i. For all cycles $c \in C$, query all edges.
   ii. If, for all $c \in C$, all edges of c exist, augment the solution: $B_{r+1} \leftarrow (B_r \setminus D) \cup C$.
c. $A_{r+1} \leftarrow A_r \setminus \{c \mid$ One or more edges in cycle c failed to exist$\}$.

3. Return B_R.

Similar to our matching results, for any element $v \in V$, the number of cycles in B_R that v belongs to and are queried is at most R. Indeed, in each of the R rounds, Algorithm 4 issues queries to vertex-disjoint augmenting structures—vertex disjoint set of cycles—and each such structure includes at most one cycle that uses vertex v.

Let us first introduce some notation that is helpful in proving Theorem 4. For a given r, we use the notation $\mathbb{E}_{Q_r}[X]$ to denote $\mathbb{E}_{Q_r}[X \mid Q_1, \ldots, Q_{r-1}]$, that is, the expected value of X, where the expectation is taken over the outcomes of Q_r when outcomes of $Q_1, \ldots, Q_{r-1}$ are fixed based on queries in the first $r-1$ rounds. Similarly, we use the notation $\mathbb{E}_{Q_r,\ldots,Q_R}[X]$ to denote $\mathbb{E}_{Q_r,\ldots,Q_R}[X \mid Q_1, \ldots, Q_{r-1}]$. Moreover, we denote by $\overline{K}(A \mid Q_1, \ldots, Q_{r-1})$ the expected size of the largest cardinality cycle-packing for (V, A) given the result of queries in $Q_1, \ldots, Q_{r-1}$.

Lemma 8. *For every $r \in [R]$, outcome of queries $Q_1, \ldots, Q_{r-1}$, and the corresponding B_{r-1}, we have*

$$\mathbb{E}_{Q_r}[|B_r|] \geq (1-\gamma)|B_{r-1}| + \gamma\left(\frac{2}{k} - \eta\right)\overline{K}(A \mid Q_1, \ldots, Q_{r-1}),$$

where $\gamma = \frac{p^{k \cdot s_{\eta,k}}}{(\frac{2}{k}-\eta)\cdot k \cdot s_{\eta,k}}$.

Proof. By Lemma 7, Q_r is a collection of at least $\frac{1}{k \cdot s_{\eta,k}}\left(K(A_r) - \frac{|B_{r-1}|}{\frac{2}{k}-\eta}\right)$ augmenting structures of size at most $s_{\eta,k}$ whose cycles are all mutually vertex-disjoint. Note that at step 2c of round $1, \ldots, r-1$, we remove any cycles that had an edge that was queried and did not exist. Therefore, all cycles that appear in the augmenting structures in Q_r consist of edges that either have never been queried or have been queried and exist. Therefore, each augmenting structure exists with probability at least $p^{k \cdot s_{\eta,k}}$. So, conditioned on $Q_1, \ldots, Q_{r-1}$, the expected increase in the size of the solution at step 2b is

$$\begin{aligned}
\mathbb{E}_{Q_r}[|B_r|] - |B_{r-1}| &\geq p^{k \cdot s_{\eta,k}}|Q_r| \\
&\geq \frac{p^{k \cdot s_{\eta,k}}}{k \cdot s_{\eta,k}}\left(|K(A_r)| - \frac{|B_{r-1}|}{\frac{2}{k}-\eta}\right) \\
&\geq \gamma\left(\left(\frac{2}{k}-\eta\right)|K(A_r)| - |B_{r-1}|\right) \\
&\geq \gamma\left(\left(\frac{2}{k}-\eta\right)\overline{K}(A \mid Q_1, \ldots, Q_{r-1}) - |B_{r-1}|\right),
\end{aligned}$$

where the last inequality follows by the fact that $|K(A_r)| \geq \overline{K}(A \mid Q_1, \ldots, Q_{r-1})$. Rearranging this proves the claim. □

We are now ready to prove Theorem 4.

Proof of Theorem 4. Let us start with a technical observation, that, for every r,

$$\mathbb{E}_{Q_{r-1}}\left[\overline{K}(A \mid Q_1, \ldots, Q_{r-1})\right] = \overline{K}(A \mid Q_1, \ldots, Q_{r-2}). \quad (5)$$

Using Lemma 7 on the Rth step of Algorithm 4 and conditioning on $Q_1, \ldots, Q_{R-1}$, we have that

$$\begin{aligned}
\mathbb{E}_{Q_R}[|B_R|] \geq (1-\gamma)|B_{R-1}| + \gamma\left(\frac{2}{k} - \eta\right) \\
\cdot \overline{K}(A \mid Q_1, \ldots, Q_{R-1}).
\end{aligned}$$

Taking expectation over Q_{R-1} in this inequality, we have

$$\begin{aligned}
&\mathbb{E}_{Q_{R-1},Q_R}[|B_R|] \\
&\geq (1-\gamma)\mathbb{E}_{Q_{R-1}}[|B_{R-1}|] \\
&\quad + \gamma\left(\frac{2}{k} - \eta\right)\mathbb{E}_{Q_{R-1}}\left[\overline{K}(A \mid Q_1, \ldots, Q_{R-1})\right] \\
&\geq (1-\gamma)\mathbb{E}_{Q_{R-1}}[|B_{R-1}|] + \gamma\left(\frac{2}{k} - \eta\right) \\
&\quad \cdot \overline{K}(A \mid Q_1, \ldots, Q_{R-2}) \\
&\geq (1-\gamma)\left((1-\gamma)|B_{R-2}| + \gamma\left(\frac{2}{k} - \eta\right)\right. \\
&\quad \left. \cdot \overline{K}(A|Q_1, \ldots, Q_{R-2})\right) + \gamma\left(\frac{2}{k} - \eta\right)\overline{K}(A \mid Q_1, \ldots, Q_{R-2}) \\
&\geq (1-\gamma)^2|B_{R-2}| + \gamma\left(\frac{2}{k} - \eta\right)(1 + (1-\gamma)) \\
&\quad \cdot \overline{K}(A|Q_1, \ldots, Q_{R-2}),
\end{aligned}$$

where the second transition is by Equation (5), and the third transition is due to applying Lemma 7 on the $(R-1)$th step. Repeating these steps by sequentially taking expectation over Q_{R-2} through Q_1 and applying Lemma 7 and Equation (5) at each step, we have

$$\begin{aligned}
\mathbb{E}_{Q_1,\ldots,Q_R}[|B_R|] &\geq (1-\gamma)^R|B_0| + \gamma\left(\frac{2}{k} - \eta\right) \\
&\quad \cdot \left(1 + (1-\gamma) + \ldots, (1-\gamma)^{R-1}\right)\overline{K}(A) \\
&\geq \gamma\left(\frac{2}{k} - \eta\right)\left(1 - (1-\gamma)^R\right)\overline{K}(A).
\end{aligned}$$

Note that, when $\eta = \frac{\epsilon}{k}$ and $R = \frac{1}{\gamma}\log(\frac{2}{\epsilon}) = \frac{(\frac{2}{k}-\eta)k \cdot s_{\eta,k}}{p^{k \cdot s_{\eta,k}}}\log(\frac{2}{\epsilon})$, we have

$$\begin{aligned}
\gamma\left(\frac{2}{k} - \eta\right)\left(1 - (1-\gamma)^R\right) &\geq \frac{2}{k}\left(1 - \frac{\eta k}{2}\right)\left(1 - (1-\gamma)^R\right) \\
&\geq \frac{2}{k}\left(1 - \frac{\epsilon}{2}\right)\left(1 - \frac{\epsilon}{2}\right) \geq \frac{2}{k}(1-\epsilon).
\end{aligned}$$

Therefore, $\mathbb{E}_{Q_1,\ldots,Q_R}[|B_R|] \geq \frac{2}{k}(1-\epsilon)\overline{K}(A)$. We complete the proof by noting that, because every cycle in A (and by extension B_R) has between two and k

vertices, the resulting approximation ratio for the optimal number of vertices covered is

$$\mathbb{E}_{Q_1,\ldots,Q_R}\left[\sum_{c\in B_R}|c|\right] \geq \frac{4}{k^2}(1-\epsilon)\overline{V}(A). \quad \square$$

8. Experimental Results on Kidney Exchange Compatibility Graphs

Our theoretical results show that our adaptive and nonadaptive algorithms recover $(1-\epsilon)$ and $(\frac{1}{2}-\epsilon)$ fractions of the omniscient optimum matching using $R = O_{\epsilon,p}(1)$ queries per vertex, respectively. Although R is a constant regardless of the number of vertices, its dependence on ϵ and p may lead to values of R that are impractical for a kidney exchange platform. To bridge this gap, we use empirical simulations from two kidney exchange compatibility graph distributions to show that our algorithms perform well in practice even for a number of per-vertex queries that is as low as $R \leq 5$.

The first distribution, from Saidman et al. (2006), was designed to mimic the characteristics of a nationwide exchange in the United States in steady state. Fielded kidney exchanges have not yet reached that point, however; with this in mind, we also include results on *real* kidney exchange compatibility graphs drawn from the first 169 match runs of the UNOS nationwide kidney exchange. Although these two families of graphs differ substantially, we find that even a small number R of nonadaptive rounds followed by a single period during which only those edges selected during the R rounds are queried, results in large gains relative to the omniscient matching.

As is common in the kidney exchange literature, in the rest of this section we loosely use the term "matching" to refer to both two-cycle packing (equivalent to the traditional definition of matching, in which two vertices connected by directed edges are translated to two vertices connected by a single undirected edge) and k-cycle packing, possibly with the inclusion of altruist-initiated chains.

This section does not directly test the algorithms presented in this paper. For the two-cycles-only case, we do directly implement Algorithm 2. However, for the cases involving longer cycles and/or chains, we do not restrict ourselves to polynomial-time algorithms (unlike in the theoretical part of this paper), instead choosing to optimally solve matching problems using integer programming during each round as well as for the final matching and for the omniscient benchmark matching. This decision is informed by the current practice in kidney exchange, in which computational resources are much less of a problem than human or monetary resources—of which the latter two are necessary for querying edges.

In our experiments, the planning of which edges to query proceeds in rounds as follows. Each round of matching calls as a subsolver the matching algorithm presented by Dickerson et al. (2019), which includes edge failure probabilities in the optimization objective to provide a maximum-discounted-utility matching. The set of cycles and chains present in a round's discounted matching are added to a set of edges to query, and then those cycles and chains are constrained from appearing in future rounds. After all rounds are completed, this set of edges is queried, and a final maximum discounted utility matching is compared against an omniscient matching that knows the set of nonfailing edges up front.

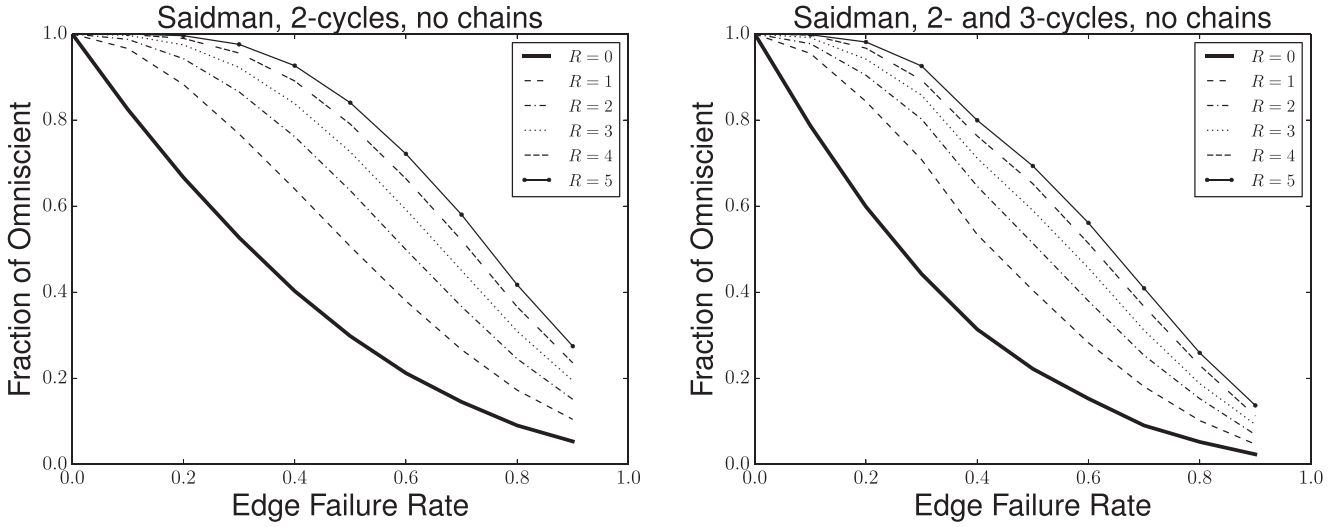
8.1. Experiments on Dense Generated Graphs from Saidman et al. (2006)

We begin by looking at graphs drawn from a distribution from Saidman et al. (2006), hereafter referred to as "the Saidman generator." This generator takes into account the blood types of patients and donors (such that the distribution is drawn from the general U.S. population) as well as three levels of PRA and various other medical characteristics of patients and donors that may affect the existence of an edge. Fielded kidney exchanges currently do not uniformly sample their pairs from the set of all needy patients and able donors in the United States as assumed by the Saidman generator; rather, exchanges tend to get hard-to-match patients who have not received an organ through other means. Because of this, the Saidman generator tends to produce compatibility graphs that are significantly denser than those seen in fielded kidney exchanges today (see, e.g., Ashlagi et al. (2011a, 2013)).

Figure 4 presents the fraction of the omniscient objective achieved by $R \in \{0, 1, \ldots, 5\}$ nonadaptive rounds of edge testing for generated graphs with 250 patient–donor pairs and no altruistic donors, constrained to two-cycles only (left) and both two- and three-cycles (right). Note that the case $R = 0$ corresponds to no edge testing, in which a maximum discounted utility matching is determined by the optimization method of Dickerson et al. (2019) and then compared directly to the omniscient matching. The x-axis varies the uniform edge failure rate f from 0.0, at which edges do not fail, to 0.9, at which edges only succeed with a 10% probability. Given an edge failure rate of f in the following figures, we can translate to the p used in the theoretical section of the paper as follows: a two-cycle in a matching represents both directions of an edge and, therefore, exists with probability $p_{2\text{-cycle}} = (1-f)^2$, and an edge in a three-cycle packing only represents a single direction of compatibility and exists with probability $p_{3\text{-cycle}} = 1-f$, and a three-cycle exists with probability $(1-f)^3$.

The utility of even a small number of edge queries is evident in Figure 4. Just a single round of testing

Figure 4. Saidman Generator Graphs Constrained to Two-Cycles Only (Left) and Both Two- and Three-Cycles (Right)



($R = 1$) results in 50.6% of omniscient—compared with just 29.8% with no edge testing—for edge failure probability $f = 0.5$ in the two-cycle case, and there are similar gains in the two- and three-cycle case. For the same failure rate, setting $R = 5$ captures 84.0% of the omniscient two-cycle matching and 69.3% in the two- and three-cycle case compared with just 22.2% when no edges are queried. Interestingly, we found no statistical difference between nonadaptive and adaptive matching on these graphs.

8.2. Experiments on Real Match Runs from the UNOS Nationwide Kidney Exchange

We now analyze the effect of querying a small number of edges per vertex on graphs drawn from the real world. Specifically, we use the first 169 match runs of the UNOS nationwide kidney exchange, which began matching in October 2010 on a monthly basis and now includes 153 transplant centers—that is, 66% of the centers in the United States—and performs match runs twice per week. These graphs, as with other fielded kidney exchanges (Ashlagi et al. 2013), are substantially less dense than those produced by the Saidman generator. This disparity between generated and real graphs has led to different theoretical results (e.g., efficient matching does not require long chains in a deterministic dense model (Dickerson et al. 2012b, Ashlagi and Roth 2014) but does in a sparse model (Ashlagi et al. 2011a, Ding et al. 2015) and empirical results (both in terms of match composition and experimental tractability (Constantino et al. 2013, Glorie et al. 2014, Anderson et al. 2015b) in the past—a trend that continues here.

Figure 5 shows the fraction of the omniscient two-cycle and two-cycle with chains match size achieved

Figure 5. Real UNOS Match Runs Constrained to Two-Cycles (Left) and Both Two-Cycles and Chains (Right)

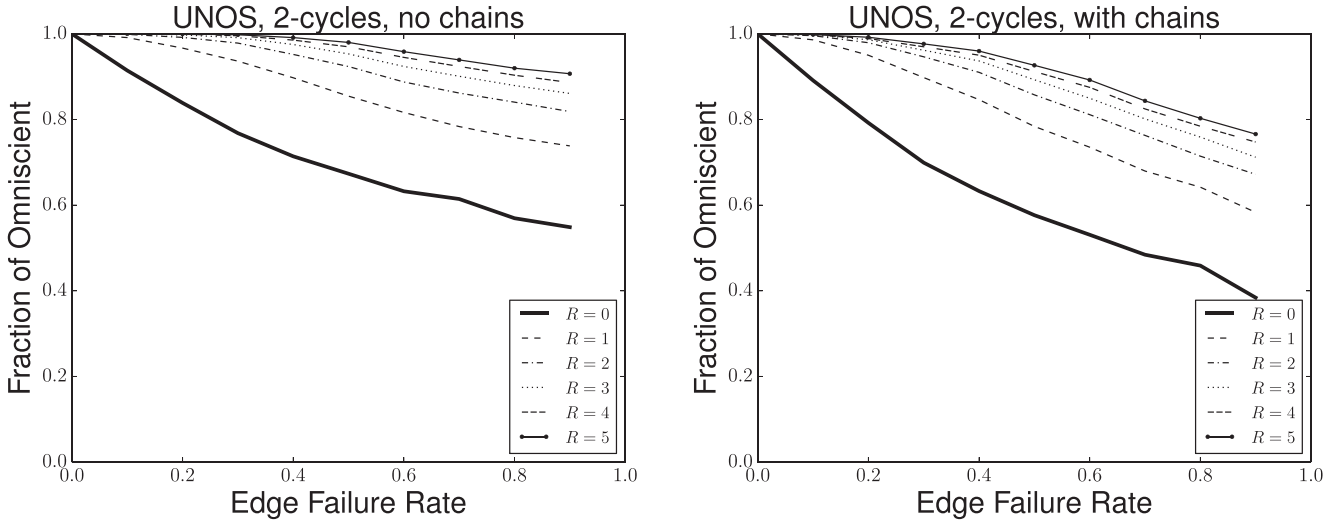
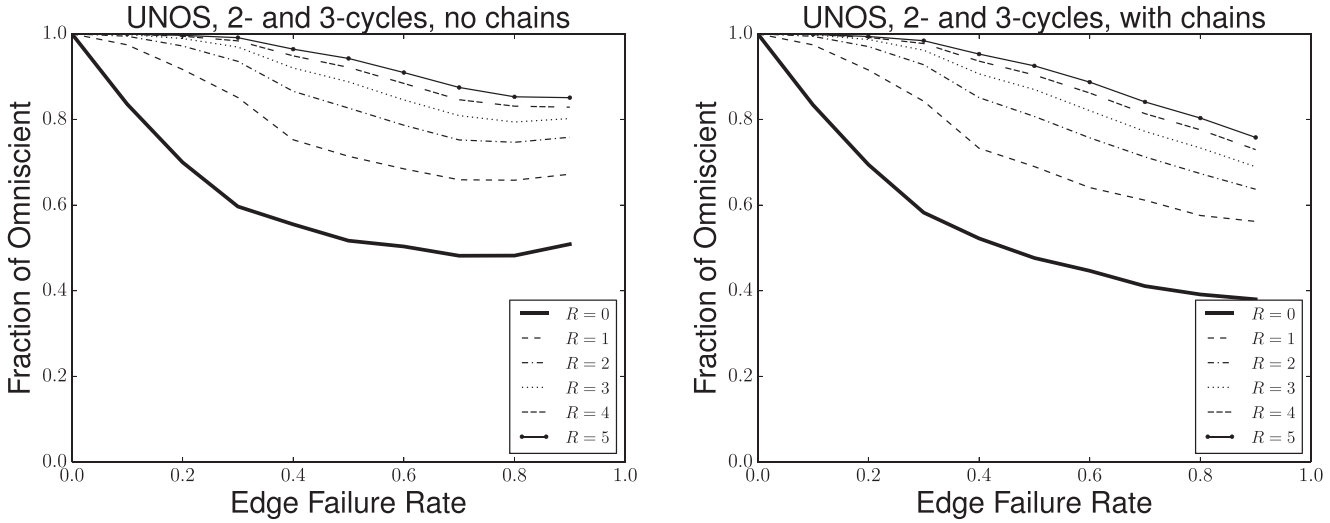


Figure 6. Real UNOS Match Runs with Two- and Three-Cycles and No Chains (Left) and with Chains (Right)



by using only two-cycles or both two-cycles and chains and some small number of nonadaptive edge query rounds $R \in \{0, 1, \ldots, 5\}$. For each of the 169 pretest compatibility graphs and each of the edge failure rates, 50 different ground truth compatibility graphs were generated. Chains can partially execute; that is, if the third edge in a chain of length three fails, then we include all successful edges (in this case, two edges) until that point in the final matching. More of the omniscient matching is achieved (even for the $R = 0$ case) on these real-world graphs than on those from the Saidman generator presented in Section 8.1. Still, the gain realized even by a small number of edge query rounds is stark with $R = 5$ achieving more than 90% of the omniscient objective for every failure rate in the two-cycles-only case and more than 75% of the omniscient objective when chains are included (and typically much more).

Figure 6 expands these results to the case with two- and three-cycles, both without and with chains. Slightly less of the omniscient matching objective is achieved across the board, but the overall increases resulting from $R \in \{1, \ldots, 5\}$ nonadaptive rounds of testing is once again prominent. Interestingly, we did not see a significant difference in results for adaptive and nonadaptive edge testing on the UNOS family of graphs either.

Next we consider these experiments again, only this time including in the analysis empty omniscient matchings. If an omniscient matching is empty, then our algorithm achieves at most zero matches as well. Previously, we removed these cases from the

Figure 7. Real UNOS Match Runs, Restricted Matching of Two-Cycles Only Without Chains (Left) and with Chains (Right), Including Zero-Sized Omniscient Matchings

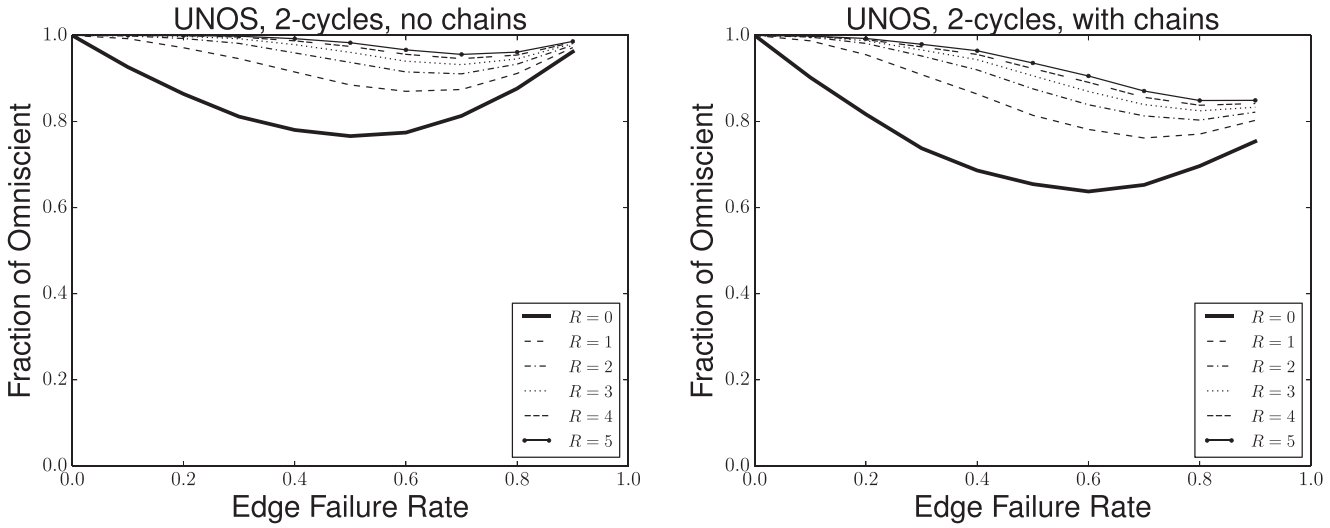
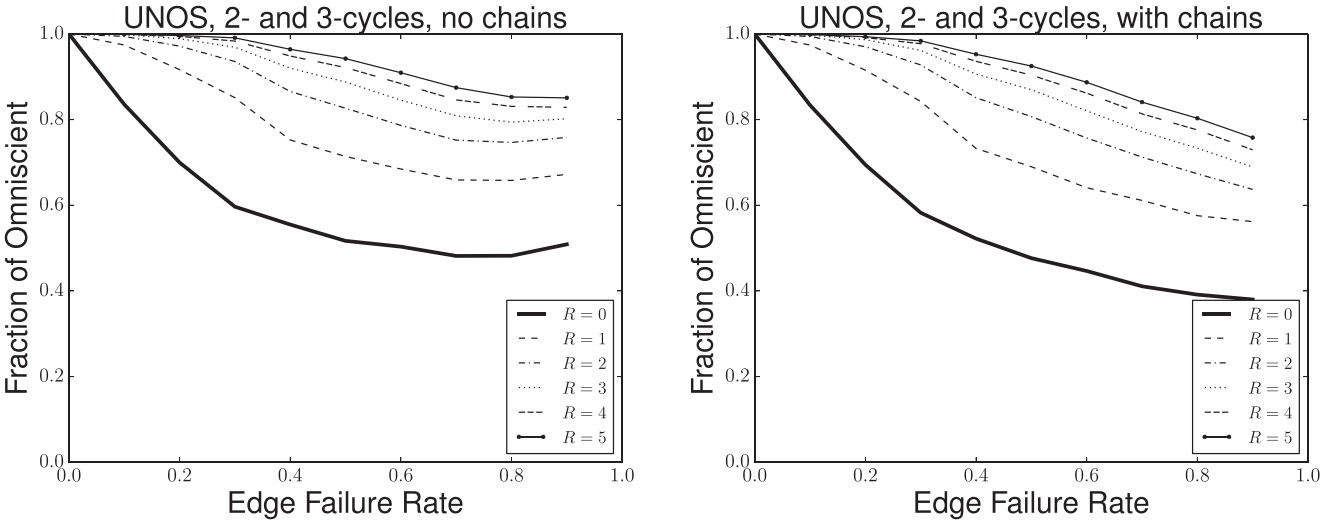


Figure 8. Real UNOS Match Runs, Matching with Two- and Three-Cycles Without Chains (Left) and with Chains (Right), Including Zero-Sized Omniscient Matchings



experimental analysis because achieving zero matches (using any method) out of zero possible matches trivially achieves 100% of the omniscient matching; by not including those cases, we provided a more conservative experimental analysis. Here, we include those cases and rerun the analysis.

Figure 7 mimics Figure 5. It shows results for two-cycle matching on the UNOS compatibility graphs without chains (left) and with chains (right) for $R \in \{0, 1, \ldots, 5\}$ and varying levels of $f \in \{0, 0.1, \ldots, 0.9\}$. We witness a marked increase in the fraction of omniscient matching achieved as f gets close to 0.9; this is due to the relatively sparse UNOS graphs admitting no matchings for high failure rates.

Figure 8 shows the same experiments as Figure 7, only this time allowing both two- and three-cycles without (left) and with (right) chains. It corresponds to Figure 6 and exhibits similar but weaker behavior to Figure 7 for high failure rates. This demonstrates the power of including three-cycles in the matching algorithm; we see that far fewer compatibility graphs admit no matchings under this less-restrictive matching policy.

Code to replicate all experiments is available at https://github.com/JohnDickerson/KidneyExchange; this code base includes graph generators but, because of privacy concerns, does not include the real match runs from the UNOS exchange.

9. Discussion and Future Research

In this paper, we addressed stochastic matching and its generalization to k-cycle packing from both a theoretical and experimental point of view. For the stochastic matching problem, we designed an adaptive algorithm that queries only a constant number of edges per vertex and achieves a $(1 - \epsilon)$ fraction of the omniscient solution for an arbitrarily small $\epsilon > 0$ and performs the queries in only a constant number of rounds. We complemented this result with a nonadaptive algorithm that achieves a $(0.5 - \epsilon)$ fraction of the omniscient optimum.

We then extended our results to the more general problem of stochastic k-cycle packing by designing an adaptive algorithm that achieves a $(\frac{2}{k} - \epsilon)$ fraction (respectively, $(\frac{4}{k^2} - \epsilon)$ fraction) of the cardinality of (respectively, number of vertices covered in) the omniscient optimal solution, again with only $O(1)$ queries per element. This guarantee is quite close to the best known polynomial-time approximation ratio of $\frac{3}{k+1} - \epsilon$ for the cardinality of the optimal k-cycle packing in the standard *nonstochastic* setting (Fürer and Yu 2014).

We adapted these algorithms to the kidney exchange problem and, on both generated and real data from the first 169 runs of the UNOS U.S. nationwide kidney exchange, explored the effect of a small number of edge query rounds on matching performance. In both cases—but especially on the real data—a very small number of nonadaptive edge queries per donor–patient pair results in large gains in expected successful matches across a wide range of edge failure probabilities.

In the theoretical part of this paper, we considered a setting in which every edge e exists with probability $p_e \geq p$ for a constant value of p and gave algorithms that queried a number of edges that increased as $p \to 0$. In kidney exchange, however, a small fraction of patients may be highly sensitized; there is a low probability that their crossmatch test with potentially compatible donors would be successful. This

gives rise to compatibility graphs that include a small fraction of edges with success probability $p_e \approx 0$. In this case, setting $p = \min_{e \in E} p_e$ in our theoretical results would require a large number of tests per vertex, leading to an impractical algorithm that tests every edge and finds the omniscient optimal solution. To avoid this problem, we note that, when the number of edges with small p_e is small, ignoring them affects the size of the omniscient optimal solution only to a small degree. This allows us to essentially recover our theoretical approximation guarantees. Of course, in practice, these edges are not ignored; rather they may receive specialized treatments, hence allowing for an even better outcome.

A related theoretical problem is the stochastic k-set packing problem. In this setting, we are given a collection of sets, each with cardinality at most k. Each set s exists with some known probability p_s independently of others, and we need to query the sets to find whether they exist. The objective is to output a collection of disjoint sets of maximum cardinality. Note that stochastic matching is a special case of stochastic k-set packing: each set (which corresponds to an edge) has cardinality two, that is, $k = 2$. However, the existence of different k-cycles is correlated for $k \geq 3$, and therefore k-cycle packing problems are different from k-set packing problems. Yet it is not hard to see that, similarly to our extension to k-cycle packing, our adaptive and nonadaptive stochastic matching algorithms also extend to the case of k-set packing. We refer interested readers to the preliminary version of this paper (Blum et al. 2015) for more details on the stochastic k-set packing problem.

9.1. Open Theoretical Problems

Three main open theoretical problems remain. First, our adaptive algorithm for the matching setting achieves a $(1-\epsilon)$-approximation in $O(1)$ rounds and using $O(1)$ queries per vertex. Is there a nonadaptive algorithm that achieves the same guarantee? Such an algorithm would make the practical message of the theoretical results even more appealing: instead of changing the status quo in two ways—more rounds of crossmatch tests, more tests per patient—we would only need to change it in the latter way.

Second, for the case of optimal cardinality k-cycle packing, we achieve a $(\frac{2}{k}-\epsilon)$-approximation using $O(n)$ queries—in polynomial time. In kidney exchange, however, our scarcest resource is the ability to query edges. In particular, computational hardness is circumvented in many cases through integer programming techniques (Abraham et al. 2007, Constantino et al. 2013, Dickerson et al. 2016). Therefore, it would be interesting to see if there is an exponential-time adaptive algorithm for k-cycle packing that requires $O(1)$ rounds and $O(n)$ queries and achieves $(1-\epsilon)$-approximation to the omniscient optimum. A positive answer would require a new approach because ours is inherently constrained to constant-size augmenting structures, which cannot yield an approximation ratio better than $\frac{2}{k}-\epsilon$ even if we could compute optimal solutions to k-cycle packing (Hurkens and Schrijver 1989).

Third, although we provided an adaptive algorithm for stochastic k-cycle packing for the general case of $k \geq 2$, our nonadaptive results were restricted to the case of $k = 2$. A natural question is whether there exists a nonadaptive algorithm with a good approximation guarantee for stochastic k-cycle packing when $k > 2$. One of the key ingredients in the analysis of our nonadaptive stochastic matching algorithm was to show that the benefit we drew from a new matching (whose edges were to be tested at the end of the algorithm) was (1) a large fraction of the size of the matching in the remaining graph and (2) independent of the outcome of the queries in the earlier matchings (see Equation (4)). For the case of $k = 2$, these properties hold because the optimal matching in the remaining graph at step r is a large fraction of the total matching, and the matching at step r shares no edges with prior matchings. For the case of $k > 2$, however, to assure that property (2) holds, we need to choose a k-cycle packing at step r using only those cycles that share no edges with the k-cycle packings in steps $1, \ldots, r-1$. Therefore, at every step, we need to remove from consideration all cycles that share an edge with an earlier packing. However, doing so results in a graph that has a small (or no) cycle packing, invalidating property (1). We conclude that new algorithms and techniques may be needed for the nonadaptive stochastic k-cycle packing problem.

9.2. Discussion of Policy Implications of Our Experimental Results

Policy decisions in kidney exchange have been linked to economic and computational studies since before the first large-scale exchange was fielded in 2003–2004 (Roth et al. 2004, 2005). A feedback loop exists between the reality of fielded exchanges—now not only in the United States but internationally as well—and the theoretical and empirical models that inform their operation, such that the latter has grown substantially closer to accurately representing the former in recent years. That said, many gaps still exist between the mathematical models used in kidney exchange studies and the systems that actually provide matches on a day-to-day basis.

Better approaches are often not adopted quickly, if at all, by exchanges. One reason for this is complexity—and not in the computational sense. Humans—doctors, lawyers, and other policy makers who are not

necessarily versed in optimization, economics, or computer science—and the organizations they represent understandably wish to understand the workings of an exchange's matching policy. The techniques described in this paper are particularly exciting in that they are quite easy to explain in accessible language, and they involve only mild changes to the status quo. At a high level, we are proposing to test some small number of promising potential matches for some subset of patient–donor pairs in a pool. As Section 8.2 shows, even a *single* extra edge test per pair will produce substantially better results.

Any new policy for kidney exchange has to address three practical restrictions in this space: (i) the monetary cost of crossmatches, (ii) the number of crossmatches that can be performed per person as there is an inherent limit on the amount of blood that can be drawn from a person, and (iii) the time it takes to find the matches as time plays a major role in the health of patients and crossmatches become less accurate as time passes and the results become stale. For both our nonadaptive and adaptive algorithms, even a very small number of rounds ($R \leq 5$) results in a very large gain in the objective. This is easily within the limits of considerations (i) and (ii). Our nonadaptive algorithm performs all chosen crossmatches in parallel, so the time taken by this method is similar to the current approach. Our adaptive algorithm, in practice, can be implemented by a one-time retrieval of R rounds worth of blood from each donor–patient pair, then sending that blood to a central laboratory. Most crossmatches are performed via an "immediate spin" in which the bloods are mixed together and either coagulate (which is bad) or do not (which is good). These tests are very fast, so a small number of rounds could be performed in a single day (assuming that tests in the same round are performed in parallel). Therefore, the timing constraint (iii) is not an issue for small R (such as that used in our experiments) for the adaptive algorithm.

More extensive studies would need to be undertaken before an exact policy recommendation can be made. These studies could take factors such as the monetary cost of an extra crossmatch test or variability in testing prowess across different medical laboratories into account explicitly during the optimization process. Various prioritization schemes could also be implemented to help, for example, hard-to-match pairs find a feasible match by assigning them a higher edge query budget than easier-to-match pairs. Moreover, there is a need for a closer look at other uncertainties in kidney exchange, such as the dynamic nature of participation of donors and patients and how they interact with our proposed algorithms. But the positive theoretical results presented in this paper, combined with the promising experimental results on real data, provide a firm basis and motivation for this type of policy analysis in the future.

Acknowledgments

This work was done while N. Haghtalab was at Carnegie Mellon University. A. Sharma is currently working at Google Inc., Mountain View, CA.

References

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

Adamczyk M (2011) Improved analysis of the greedy algorithm for stochastic matching. *Inform. Processing Lett.* 111(15):731–737.

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

Anderson R, Ashlagi I, Gamarnik D, Kanoria Y (2015a) A dynamic model of barter exchange. *Proc. 26th Annual ACM-SIAM Sympos. Discrete Algorithms (SODA)* (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.

Asadpour A, Nazerzadeh H, Saberi A (2008) Stochastic submodular maximization. Papadimitriou C, Zhang S, eds. *Proc. 4th Internat. Workshop Internet Network Econom. (WINE)* (Springer, Berlin, Heidelberg), 477–489.

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

Ashlagi I, Jaillet P, Manshadi VH (2013) Kidney exchange in dynamic sparse heterogenous pools. *Proc. 14th ACM Conf. Electronic Commerce (EC)* (ACM, New York), 25–26.

Ashlagi I, Gamarnik D, Rees MA, Roth AE (2011a) The need for (long) chains in kidney exchange. NBER Working Paper No. w18202, National Bureau of Economic Research, Cambridge, MA.

Ashlagi I, Gilchrist DS, Roth AE, Rees M (2011b) 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. 17th ACM Conf. Electronic Commerce (EC)* (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)* (Morgan Kaufmann Publishers Inc., San Francisco), 405–411.

Bansal N, Gupta A, Li J, Mestre J, Nagarajan V, Rudra A (2012) When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica* 63(4):733–762.

Blum A, Gupta A, Procaccia AD, Sharma A (2013) Harnessing the power of two crossmatches. *Proc. 14th ACM Conf. Electronic Commerce (EC)* (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. 16th ACM Conf. Electronic Commerce (EC)* (ACM, New York), 325–342.

Bollobás, B (2001) *Random Graphs*, 2nd ed. (Cambridge University Press, Cambridge, UK).

Chen N, Immorlica N, Karlin AR, Mahdian M, Rudra A (2009) Approximating matches made in heaven. Albers S, Marchetti-Spaccamela A, Matias Y, Nikoletseas S, Thomas W, eds. *Proc. 36th Internat. Colloquium Automata, Languages Programming (ICALP)* (Springer, Berlin, Heidelberg), 266–278.

Constantino M, Klimentova X, Viana A, Rais A (2013a) New insights on integer-programming models for the kidney exchange problem. *Eur. J. Oper. Res.* 231(1):57–68.

Costello KP, Tetali P, Tripathi P (2012) Matching with commitment. Czumaj A, Mehlhorn K, Pitts A, Wattenhofer R, eds. *Proc. 39th*

Internat. Colloquium Automata, Languages Programming (ICALP) (Springer, Berlin, Heidelberg), 822–833.

Dean BC, Goemans MX, Vondrak J (2004) Approximating the stochastic knapsack problem: The benefit of adaptivity. *Math. Oper. Res.* 33(4):945–964.

Dickerson JP, Sandholm T (2015) FutureMatch: Combining human value judgments and machine learning to match in dynamic environments. *Proc. 29th AAAI Conf. Artificial Intelligence (AAAI), Austin, TX*, 622–628.

Dickerson JP, Procaccia AD, Sandholm T (2012a) Dynamic matching via weighted myopia with application to kidney exchange. *Proc. 26th AAAI Conf. Artificial Intelligence (AAAI), Toronto, Canada*, 1340–1346.

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

Dickerson JP, Procaccia AD, Sandholm T (2019) Failure-aware kidney exchange. *Management Sci.* 65(4):1768–1791.

Dickerson JP, Manlove DF, Plaut B, Sandholm T, Trimble J (2016) Position-indexed formulations for kidney exchange. *Proc. 17th ACM Conf. Electronic Commerce (EC)* (ACM, New York), 25–42.

Ding Y, Ge D, He S, Ryan C (2015) A non-asymptotic approach to analyzing kidney exchange graphs. *Proc. 16th ACM Conf. Electronic Commerce (EC)* (ACM, New York), 257–258.

Fürer M, Yu H (2014) Approximate the *k*-set packing problem by local improvements. Fouilhoux P, Gouveia L, Mahjoub A, Paschos V, eds. *Combinatorial Optimization (ISCO 2014)*, Lecture Notes in Computer Science, vol. 8596 (Springer, Cham, Switzerland).

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. *Proc. 53rd Sympos. Foundations Comput. Sci. (FOCS)* (IEEE, Piscataway, NJ), 718–727.

Gupta A, Nagarajan V (2013) A stochastic probing problem with applications. Goemans M, Correa J, eds. *Proc. 16th Conf. Integer Programming Combinatorial Optim. (IPCO)* (Springer, Berlin, Heidelberg), 205–216.

Gupta A, Krishnaswamy R, Nagarajan V, Ravi R (2012) Approximation algorithms for stochastic orienteering. *Proc. 23rd Annual ACM-SIAM Sympos. Discrete Algorithms (SODA)* (SIAM, Philadelphia), 1522–1538.

Hurkens CAJ, Schrijver A (1989) On the size of systems of sets every *t* of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discrete Math.* 2(1):68–72.

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. *American Transplant Congress (ATC)*, talk abstract.

Manlove D, O'Malley G (2015) Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *ACM J. Experiment. Algorithmics* 19:2–6.

Molinaro M, Ravi R (2011) The query-commit problem. Working paper, University of California, Berkeley, Berkeley.

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

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

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

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

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

U.S. Department of Health and Human Services (2018) Organ procurement and transplantation network. Accessed June 10, 2018, http://optn.transplant.hrsa.gov.

Avrim Blum received his BS, MS, and PhD from MIT in 1987, 1989, and 1991, respectively. In 2017, he joined the Toyota Technological Institute at Chicago as chief academic officer. His main research interests are in theoretical computer science and machine learning, including machine learning theory, approximation algorithms, algorithmic game theory, and database privacy, as well as connections among them.

John P. Dickerson is an assistant professor of computer science at the University of Maryland. His research centers on solving economic problems using techniques from computer science, stochastic optimization, and machine learning. He is an NSF CAREER awardee, Facebook Fellow, NDSEG Fellow, and Siebel Scholar.

Nika Haghtalab is a postdoctoral researcher at Microsoft Research, New England. Her research is on the theoretical aspects of machine learning and algorithmic economics with a focus on developing foundations for machine learning that accounts for social and economic interactions with people. Her distinctions include the CMU School of Computer Science Dissertation award (2018), a Microsoft Research fellowship, a Facebook fellowhsip, an IBM fellowship, and a Siebel scholarship.

Ariel Procaccia is an associate professor in the computer science department at Carnegie Mellon University. He usually works on problems at the interface of AI, theoretical computer science, and economics. His distinctions include a Guggenheim Fellowship (2018), the IJCAI Computers and Thought Award (2015), a Sloan Research Fellowship (2015), an NSF CAREER Award (2014), and the IFAAMAS Victor Lesser Distinguished Dissertation Award (2009).

Tuomas Sandholm is Angel Jordan Professor of Computer Science at Carnegie Mellon. He is co-director of CMU AI and director of the Electronic Marketplaces Lab. He is the founder and CEO of Optimized Markets, Strategic Machine, and Strategy Robot. His honors include the Minsky Medal, Computers and Thought Award, ACM Autonomous Agents Research Award, Allen Newell Award for Research Excellence, Sloan Fellowship, Edelman Laureateship, and NSF CAREER Award. He is fellow of INFORMS, ACM, and AAAI.

Ankit Sharma is a software engineer at Google, Inc. His work helps people discover the best articles on the internet for their hobbies and interests. His research interests are in approximation algorithms and algorithmic game theory. He received his PhD from Carnegie Mellon University, advised by Avrim Blum and Anupam Gupta. His thesis focused on designing novel ways of allocating limited resources under a variety of constraints and objectives.