X

Ignorance is Almost Bliss: Near-Optimal Stochastic Matching With Few Queries

AVRIM BLUM, Carnegie Mellon University
JOHN P. DICKERSON, Carnegie Mellon University
NIKA HAGHTALAB, Carnegie Mellon University
ARIEL D. PROCACCIA, Carnegie Mellon University
TUOMAS SANDHOLM, Carnegie Mellon University
ANKIT SHARMA, Solvvy

The stochastic matching problem deals with 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-set packing, where the problem is to find a maximum packing of sets, each of which exists with some probability. In this paper, we provide edge and set query algorithms for these two problems, respectively, that provably achieve some fraction of the omniscient optimal solution.

Our main theoretical result for the stochastic matching (i.e., 2-set packing) problem is the design of an *adaptive* algorithm that queries only a constant number of edges per vertex and achieves a $(1-\epsilon)$ fraction of the omniscient optimal solution, for an arbitrarily small $\epsilon > 0$. Moreover, this adaptive algorithm performs the queries in only a constant number of rounds. We complement this result with a *non-adaptive* (i.e., one round of queries) algorithm that achieves a $(0.5-\epsilon)$ fraction of the omniscient optimum. We also extend both our results to stochastic k-set packing by designing an adaptive algorithm that achieves a $(\frac{2}{k}-\epsilon)$ fraction of the omniscient optimal solution, again with only $O(1)$ queries per element. This guarantee is close to the best known polynomial-time approximation ratio of $\frac{3}{k+1}-\epsilon$ for the *deterministic* k-set packing problem [Fürer and Yu 2013].

We empirically explore the application of (adaptations of) these algorithms to the kidney exchange problem, where patients with end-stage renal failure swap willing but incompatible donors. We show on both generated data and on real data from the first 169 match runs of the UNOS nationwide kidney exchange that even a very small number of non-adaptive edge queries per vertex results in large gains in expected successful matches.

Categories and Subject Descriptors: J.4 [**Social and Behavioral Sciences**]: Economics

General Terms: Algorithms, Economics, Experimentation

Additional Key Words and Phrases: Stochastic matching, stochastic k-set packing, kidney exchange

1. INTRODUCTION

In the *stochastic matching* problem, we are given an undirected graph $G = (V, E)$, where 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 the 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-set packing*, where each set exists only with some probability, and the problem is to find a packing of maximum size of those sets that do exist.

Without any constraints, one can simply query all edges or sets, and then output the maximum matching or packing over those that exist—but this level of freedom may not always be available. We are interested in the tradeoff between the number of queries and the fraction of the omnsicient optimal solution achieved. Specifically, we ask: In order 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-set 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 $O(1)$ edges incident to any particular vertex, requires $O(1)$ rounds of parallel queries, and achieves a $(1 - \epsilon)$ fraction of the omniscient optimum.[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 non-adaptive 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 non-adaptive algorithm that achieves a $0.5(1 - \epsilon)$-approximation (for arbitrarily small $\epsilon > 0$) to the omniscient optimum. We extend our matching results to a more general stochastic model in Appendix C.

We extend our results to the stochastic k-set packing problem, where we are given a collection of sets, each with cardinality at most k. Stochastic Matching is a special case of Stochastic k-set packing: each set (which corresponds to an edge) has cardinality 2, that is, $k = 2$. In stochastic k-set packing, each set s exists with some known probability p_s, and we need to query the sets to find whether they exist. Our objective is to output a collection of *disjoint* sets of maximum cardinality. We present adaptive and non-adaptive polynomial-time algorithms that achieve, for any constant $\epsilon > 0$, at least $(\frac{2}{k} - \epsilon)$ and $(1 - \epsilon)\frac{(2/k)^2}{2/k+1}$ fraction, respectively, of the omniscient optimum, again using $O(1)$ queries per element and hence $O(n)$ overall. For the sake of comparison, the best known *polynomial-time* algorithm for optimizing k-set packing in the *standard non-stochastic setting* has an approximation ratio of $\frac{3}{k+1} - \epsilon$ [Fürer and Yu 2013].

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]), where the omniscient optimum is known to be close to n. But in our work we are dealing with *arbitrary* graphs where 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 3.

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 to the omniscient optimum, then the current structure must admit a *large collection of disjoint constant-sized 'augmenting' structures*. These augmenting structures are composed of sets 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-set 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 February 7, 2015, there are 101,556 people on the US national waiting list,[2] making the median

[1] This guarantee holds as long as all the non-zero p_e's are bounded away from zero by some constant. The constant can be arbitrarily small but should not depend on n.

[2] http://optn.transplant.hrsa.gov.

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*, where 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.

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* takes 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. While 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 [Dickerson et al. 2013; Leishman et al. 2013]. This means that, if we tested a perfect 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 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.*

The above discussion pertains to pairwise kidney exchange. However, modern kidney exchange programs regularly employ swaps involving three donor-patient pairs, which are known to provide significant benefit compared to pairwise swaps alone [Roth et al. 2007; Ashlagi and Roth 2014]. Mathematically, we can consider a directed graph, where an edge (u, v) means that the donor of pair u is compatible with the patient of pair v (before a crossmatch test was performed). In this graph, pairwise and 3-way exchanges correspond to 2-cycles and 3-cycles, respectively. Our adaptive algorithm for 3-set packing then provides a $(2/3)$-approximation to the omniscient optimum, using only $O(1)$ crossmatch tests per patient and $O(n)$ overall. While 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 8.1.

To bridge the gap between theory and practice, we provide experiments on both simulated data and real data from the first 169 match runs of the United Network for Organ Sharing (UNOS) US nationwide kidney exchange, which now includes 143 transplant centers—approximatey 60% of the transplant centers in the US. 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 non-adaptive 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 matching. We discuss the policy implications of this promising result in Section 8.2.

2. RELATED WORK

While 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, where the algorithm is *forced* to add any queried edge to the matching if the edge is found to exist. In the papers of Goel and Tripathi [2012] and Costello et al. [2012], lower bounds of 0.56 and 0.573, respectively, and upper bounds of 0.7916 and 0.898, respectively, are established for graphs in which no information is available about the edges, and in which each edge e exists with a given probability p_e, respectively. 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* where 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 that the algorithm is compared to switches from the omniscient optimum to the constrained optimum, i.e., 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 to 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 2, 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).

2.2. Kidney exchange

Early models of kidney exchange did not explicitly consider the possibility of a modeled edge existing only probabilistically in reality. Recent research by Dickerson et al. [2013] and Anderson et al. [2015b] focuses on the kidney exchange application and restricts attention to a single cross-match 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 3-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. 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 theoret-

ical [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 where 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.[3] 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 will assume for ease of exposition that $p_e = p$ for all edges $e \in E$. Our results hold when p is a lower bound, i.e., $p_e \geq p$ for all $e \in E$. Furthermore, in Appendix C, we show that we can extend our results to a more general setting where 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 *non-adaptive* 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 $\bigcup_{1 \leq i \leq r} X_i$. 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.

To gain some intuition about our goal of arbitrarily good approximations to the omniscient optimum, and why it is challenging, let us consider the naïve (non-adaptive) algorithm which queries $o(n)$ random neighbors of each vertex. The following example shows that this algorithm performs poorly.

Example 3.1. 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 vertices of B and C, and two complete bipartite graphs, one between A and B, and another between C and D. See Figure 1 for an illustration. Let $p = 0.5$ be the existence probability of any edge.

The omniscient optimal solution can use any edge, and, in particular, it can use the edges between B and C. Since, 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

[3] In the notation $M(E)$, we intentionally suppress the dependence on the vertex set V, since we care about the maximum matchings of different subsets of edges for a fixed vertex set.

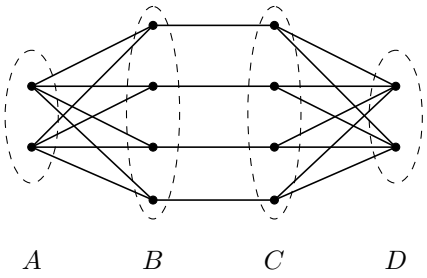

Fig. 1. Illustration of the construction in Example 3.1, for $t = 4$ and $\beta = 1/2$.

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 0, as $n \to \infty$. For $\alpha = 0.5$ and $\beta = 0.75$, the approximation of the naïve algorithm ratio is $O(1/n^{0.25})$, at best.

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

(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 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 non-existing edges.
  (d) $W_{r+1} \leftarrow W_r \cup Q''_r$.
  (e) Set M_r to the maximum matching in $\left(V, \bigcup_{j=1}^{r} Q'_j\right)$.
(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 $O_r \Delta M_{r-1}$. Since 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 4.1. *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 non-stochastic setting.)

LEMMA 4.2. *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 1, 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

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

□

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 non-existence) of the edges in $\bigcup_{i=1}^{r-1} Q_i$. We use Z_r to denote the expected size of the maximum matching in graph $G=(V,E)$ given the results of the queries $\bigcup_{i=1}^{r-1} Q_i$. More formally, $Z_r = \overline{M}(E|\bigcup_{i=1}^{r-1} Q_i)$. 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 $\mathbb{E}_{Q_r,\ldots,Q_R}[X]$ 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 4.3, 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 4.3. *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 4.2, 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 non-queried edges, cannot be larger than O_r, which is the maximum matching assuming that every non-queried edge exists. □

We are now ready to prove the theorem.

PROOF OF THEOREM 4.1. Let $L = \frac{4}{\epsilon} - 1$; it is assumed to be an odd integer for ease of exposition.[4] By Lemma 4.3, 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 will use this inequality repeatedly to derive our result. We will also require the equality

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

First, applying Lemma 4.3 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 $Q_1, \ldots, Q_{R-1}$. 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 (3), we know that $\mathbb{E}_{Q_{R-1}}[Z_R] = Z_{R-1}$. Furthermore, we can apply Lemma 4.3 to $\mathbb{E}_{Q_{R-1}}[|M_{R-1}|]$ to get the following inequality:

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

We repeat the above steps by sequentially taking expectations over Q_{R-2} through Q_1, and at each step applying Equation (3) and Lemma 4.3. This gives us

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

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

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

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

[4]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 in the appendix.

In Appendix C, we extend our results to the setting where 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, first, where an adversary chooses the vertex parameters, and second, where 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.

5. NON-ADAPTIVE 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 non-adaptive 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 non-adaptive interpretation of our adaptive approach take us to the omniscient optimum?

In more detail, our non-adaptive 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 will show that, for any arbitrarily small $\epsilon > 0$, the algorithm finds a $0.5(1-\epsilon)$-approximate solution. Since we allow an arbitrarily small (though 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 NON-ADAPTIVE ALGORITHM FOR STOCHASTIC MATCHING: 0.5-APPROXIMATION

Input: A graph $G(V, E)$.
Parameter: $R = \frac{\log(2/\epsilon)}{p^{2/\epsilon}}$
Algorithm:

(1) Initialize $W_1 \leftarrow \emptyset$.
(2) For $r = 1, \ldots, R$, do
   (a) Compute a maximum matching, O_r, in $\left(V, E \setminus \bigcup_{1 \le i \le r-1} W_i\right)$.
   (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, whose proof appears in Appendix A, establishes the approximation guarantee of Algorithm 2.

THEOREM 5.1. *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 $\overline{M}(E)$.*

As explained in Section 8.1, we do not know whether in general non-adaptive algorithms can achieve a $(1-\epsilon)$-approximation with $O(1)$ queries per vertex. However, if there is such an algorithm, it is not Algorithm 2! Indeed, the next theorem (whose proof is relegated to Appendix A) shows that

the algorithm cannot give an approximation ratio better than $5/6$ to the omniscient optimum. This fact holds even when $R = \Theta(\log n)$.

THEOREM 5.2. *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 $\frac{5}{6}\overline{M}(E) + \epsilon$.*

Despite this somewhat negative result, in Section 7, 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.

6. GENERALIZATION TO K-SET PACKING

So far we have focused on stochastic matching, for ease of exposition. But our approach directly generalizes to the k-set packing problem. Here we describe this extension for the adaptive (Section 6.1) and non-adaptive (Section 6.2) cases, and relegate the details—in particular, most proofs—to Appendix B.

Formally, a k-set packing instance (U, A) consists of a set of elements U, $|U| = n$, and a collection of subsets A, such that each subset S in A contains at most k elements of U, that is, $S \subseteq U$ and $|S| \leq k$. Given such an instance, a feasible solution is a collection of sets $B \subseteq A$ such that any two sets in B are disjoint. We use $K(A)$ to denote the largest feasible solution B.

Finding an optimal solution to the k-set packing problem is **NP**-hard (see, e.g., [Abraham et al. 2007] for the special case of k-cycle packing). Hurkens and Schrijver [1989] designed a polynomial-time local search algorithm with an approximation ratio of $(\frac{2}{k} - \eta)$, using local improvements of *constant size* that depends only on η and k. We denote this constant by $s_{\eta,k}$. More formally, consider an instance (U, A) of k-set packing and let $B \subseteq A$ be a collection of disjoint k-sets. (C, D) is said to be an *augmenting structure* for B if removing D and adding C to B increases the cardinality and maintains the disjointness of the resulting collection, i.e., if $(B \cup C) \setminus D$ is a disjoint collection of k-sets and $|(B \cup C) \setminus D| > |B|$, where $C \subseteq A$ and $D \subseteq B$.

Hurkens and Schrijver [1989] have also shown that an approximation ratio better than $2/k$ cannot be achieved with structures of constant size. While subsequent work [Fürer and Yu 2013] has improved the approximation ratio, their local search algorithm finds structures of super-constant size. This is inconsistent with our technical approach, as we need each queried structure to exist (in its entirety) with constant probability.

To be more precise, Hurkens and Schrijver [1989] prove:

LEMMA 6.1 ([HURKENS AND SCHRIJVER 1989]). *Given a collection B of disjoint sets such that $|B| < (2/k - \eta)|K(A)|$, there exists an augmenting structure (C, D) for B such that both C and D have at most $s_{\eta,k}$ sets, for a constant $s_{\eta,k}$ that depends only on η and k.*

However, we need to find *many* augmenting structures. We use Lemma 6.1 to prove:

LEMMA 6.2. *If $|B| < |K(A)|$, then there exist $\frac{1}{k\, s_{\eta,k}}(|K(A)| - \frac{|B|}{\frac{2}{k} - \eta})$ disjoint augmenting structures that augment the cardinality of B, each with size at most $s_{\eta,k}$. Moreover, this collection of augmenting structures can be found in polynomial time.*

PROOF. We prove the lemma using Algorithm 3. We claim that if we run this algorithm on the k-set packing instance (U, A) and the collection B, then it will return a collection Q of at least $T = \frac{1}{k\, s_{\eta,k}}(|K(A)| - \frac{|B|}{\frac{2}{k} - \eta})$ disjoint augmenting structures (C, D) for B. By Step 2c, we are guaranteed that Q consists of *disjoint* augmenting structures. Hence, all that is left to show is that in each of the first T iterations, at Step 2a, we are able to find a *nonempty* augmenting structure (C, D) for B.

By Lemma 6.1, we know that if at iteration t it is the case that $|B| < (\frac{2}{k} - \eta)|K(A_t)|$, then we will find an augmenting structure (C, D) of size $s_{\eta,k}$ for B. To prove that the inequality holds at

each iteration $t \leq T$, we first claim that for all t,

$$|K(A_t)| \geq |K(A)| - (t-1) \cdot k \cdot s_{\eta,k} \tag{3}$$

We prove this by induction. The claim is clearly true for the base case of $t = 1$. For the inductive step, suppose it is true for t, then we know that $|K(A_t)| \geq |K(A)| - (t-1) \cdot k \cdot s_{\eta,k}$. At iteration t, the augmenting structure (C, D) can intersect with at most $s_{\eta,k} \cdot k$ sets of $K(A_t)$. This is true since $K(A_t)$ consists of *disjoint* sets, and the augmenting structure (C, D) is of size at most $s_{\eta,k}$. Hence, Step 2c reduces $|K(A_t)|$ by at most $k \cdot s_{\eta,k}$. So, $|K(A_{t+1})| \geq |K(A_t)| - k \cdot s_{\eta,k}$. Combining the two inequalities, $|K(A_t)| \geq |K(A)| - (t-1) \cdot k \cdot s_{\eta,k}$ and $|K(A_{t+1})| \geq |K(A_t)| - k \cdot s_{\eta,k}$, we have $|K(A_{t+1})| \geq |K(A)| - t \cdot k \cdot s_{\eta,k}$. This establishes Equation (3).

We conclude that if the for-loop adds *non-empty* augmenting structures only for the first t rounds, it must be the case that $|B| \geq (\frac{2}{k} - \eta)|K(A_{t+1})|$, and therefore $|B| \geq (\frac{2}{k} - \eta)(|K(A)| - t \cdot k \cdot s_{\eta,k})$ which implies that $t \geq \frac{1}{k\, s_{\eta,k}}(|K(A)| - \frac{|B|}{\frac{2}{k} - \eta})$. □

Algorithm 3 FINDING CONSTANT-SIZE DISJOINT AUGMENTING STRUCTURES FOR k-SETS

Input: k-set packing instance (U, A) and a collection $B \subseteq A$ of disjoint sets.
Output: Collection Q of disjoint augmenting structures (C, D) for B.
Parameter: $s_{\eta,k}$ (the desired maximum size of the augmenting structures)
Algorithm:

(1) Initialize $A_1 \leftarrow A$ and $Q \leftarrow \phi$ (empty set).
(2) For $t = 1, \cdots, |A|$
  (a) Find an augmenting structure (C, D) of size $s_{\eta,k}$ for B on the k-set instance (U, A_t).
  (b) Add (C, D) to Q. (If C is an empty set, break out of the loop.)
  (c) Set A_{t+1} to be A_t minus the collection C and any set in $A_t \setminus B$ that intersects with C.
(3) Output Q.

6.1. Adaptive algorithm for k-set packing

Turning to the stochastic version of the problem, given (U, A), let A_p be a random subset of A where each set from A is included in A_p independently with probability p. We then define $\overline{K}(A)$ to be $\mathbb{E}[|K(A_p)|]$, where the expectation is taken over the random draw A_p. Similarly to the matching setting, this is the omniscient optimum—our benchmark.

We extend the ideas introduced earlier in the paper for matching, together with Lemma 6.2 and additional ingredients, to obtain the following result for the adaptive problem.

THEOREM 6.3. *There exists an adaptive polynomial-time algorithm that, given a k-set instance (U, A) and $\epsilon > 0$, uses $O(1)$ rounds and $O(n)$ queries overall, and returns a set B_R whose expected cardinality is at least a $(1 - \epsilon)\frac{2}{k}$ fraction of $\overline{K}(A)$.*

With an eye toward Theorem 6.3, Algorithm 4 is a polynomial-time algorithm that can be used to find such a packing that approximates the omniscient optimum. In each round r, the algorithm maintains a feasible k-set packing B_r based on the k-sets that have been queried so far. It then computes a collection Q_r of disjoint, small augmenting structures with respect to the current solution B_r, where the augmenting structures are composed of sets that have not been queried so far. It issues queries to these augmenting structures, and uses those that are found to exist to augment the current solution. The augmented solution is fed into the next round.

Similarly to our matching results, for any element $v \in U$, the number of sets that it belongs to and are also queried is at most R. Indeed, in each of the R rounds, Algorithm 4 issues queries to disjoint augmenting structures, and each augmenting structure includes at most one set per element.

Algorithm 4 ADAPTIVE ALGORITHM FOR STOCHASTIC k-SET PACKING

Input: A k-set instance (U, A), and $\epsilon > 0$.

Parameters: $\eta = \frac{\epsilon}{k}$ and $R = \frac{(\frac{2}{k}-\eta)\ k\ s_{\eta,k}}{p^{s_{\eta,k}}} \log(\frac{2}{\epsilon})$ (For a $(1-\epsilon)(\frac{2}{k})$-approximation)

Algorithm:

(1) Initialize $r \leftarrow 1$, $B_0 \leftarrow \emptyset$ and $A_1 \leftarrow A$.
(2) For $r = 1, \ldots, R$, do
  (a) Initialize B_r to B_{r-1}.
  (b) Let Q_r be the set of augmenting structures given by Algorithm 3 on the input consisting of the k-set packing instance (U, A_r), the collection B_r, and the parameter $s_{\eta,k}$.
  (c) For each augmenting structure $(C, D) \in Q_r$.
    i. Query all sets in C.
    ii. If all the sets of C exist, augment the current solution: $B_r \leftarrow (B_r \setminus D) \cup C$.
  (d) Set A_{r+1} to be A_r after removing queried sets that were found not to exist.
(3) Return B_R.

6.2. Non-adaptive algorithm for k-set packing

Once again, when going from the adaptive case to the non-adaptive case, the fraction of the omniscient optimum that we can obtain becomes significantly worse.

THEOREM 6.4. *There exists a non-adaptive polynomial-time algorithm that, given a k-set instance (U, A) and $\epsilon > 0$, uses $O(n)$ queries overall and returns a k-set packing with expected cardinality $(1-\epsilon)\frac{(2/k)^2}{2/k+1}\overline{K}(A)$.*

We present such a polynomial-time non-adaptive algorithm, Algorithm 5, that proceeds as follows. For R rounds, at every round, using the local improvement algorithm of Hurkens and Schrijver [1989], we find a $(\frac{2}{k}-\eta)$-approximate solution to the k-set instance and remove it. Then, we query every set that is included in these R solutions. We show that the expected cardinality of the maximum packing on the chosen sets is a $(1-\epsilon)\frac{(2/k)^2}{2/k+1}$ approximation of the expected optimal packing. As usual, it is easy to see that $O(n)$ queries are issued overall.

Algorithm 5 NON-ADAPTIVE ALGORITHM FOR STOCHASTIC k-SET PACKING

Input: A k-set packing instance (U, A), and $\epsilon > 0$.

Parameters: $\eta = \frac{\epsilon}{2k}$ and $R = \frac{(\frac{2}{k}-\eta)\ k\ s_{\eta,k}}{p^{s_{\eta,k}}} \log(\frac{2}{\epsilon})$. (For $(1-\epsilon)\frac{(2/k)^2}{2/k+1}$-approximation)

Algorithm:

(1) Let $B_0 \leftarrow \emptyset$.
(2) For $r = 1, \ldots, R$, do
  (a) $O_r \leftarrow$ a $(\frac{2}{k}-\eta)$-approximate solution to the k-set instance $(U, A \setminus \bigcup_{i=1}^{r-1} B_i)$. ($O_r$ is found using the local improvement algorithm of Hurkens and Schrijver [1989].)
  (b) Set $B_r \leftarrow B_{r-1} \cup O_r$.
(3) Query the sets in O_1, and assign Q_1 to be the sets that are found to exist.
(4) For $r = 2, \cdots, R$, do
  (a) Find augmenting structures in O_r that augment Q_{r-1}. This is achieved by giving the instance $(U, Q_{r-1} \cup O_r)$ and solution Q_{r-1} as input (with parameter $s_{\eta,k}$) to Algorithm 3.
  (b) Query all the augmenting structures in O_r, and augment Q_{r-1} with the ones that are found to exist. Call the augmented solution Q_r.
(5) Output Q_R.

Importantly, the statements of Theorems 4.1 and 5.1 are special cases of the statements of Theorems 6.3 and 6.4, respectively, for $k = 2$, although on a technical level the $k = 2$ case must be handled separately (as we did, with less cumbersome terminology and technical constructions than in the general k-set packing setting).

7. EXPERIMENTAL RESULTS ON KIDNEY EXCHANGE COMPATIBILITY GRAPHS

In this section, we support our theoretical results with empirical simulations from two kidney exchange compatibility graph distributions. The first distribution, due to 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, though; 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. While these two families of graphs differ substantially, we find that even a small number R of non-adaptive 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 will loosely use the term "matching" to refer to both 2-set packing (equivalent to the traditional definition of matching, where two vertices connected by directed edges are translated to two vertices connected by a single undirected edge) and k-set packing, possibly with the inclusion of altruist-initiated chains.

This section does not directly test the algorithms presented in this paper. For the 2-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 theory 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, where 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. [2013], 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 non-failing edges up front.

7.1. Experiments on dense generated graphs due to Saidman et al. [2006]

We begin by looking at graphs drawn from a distribution due to 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 United States 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 US, 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. 2011, 2013]).

Figure 2 presents the fraction of the omniscient objective achieved by $R \in \{0, 1, \ldots, 5\}$ non-adaptive rounds of edge testing for generated graphs with 250 patient-donor pairs and no altruistic donors, constrained to 2-cycles only (left) and both 2- and 3-cycles (right). Note that the case $R = 0$ corresponds to no edge testing, where a maximum discounted utility matching is determined by the optimizer and then compared directly to the omniscient matching. The x-axis varies the uniform

edge failure rate f from 0.0, where edges do not fail, to 0.9, where edges only succeed with a 10% probability. Given an edge failure rate of f in the figures below, we can translate to the p used in the theoretical section of the paper as follows: 2-cycles exists with probability $p_{\text{2-cycle}} = (1-f)^2$, while a 3-cycle exists with $p_{\text{3-cycle}} = (1-f)^3$. For example, in the case of $f = 0.9$, a 3-cycle exists with very low probability $p = 0.001$.

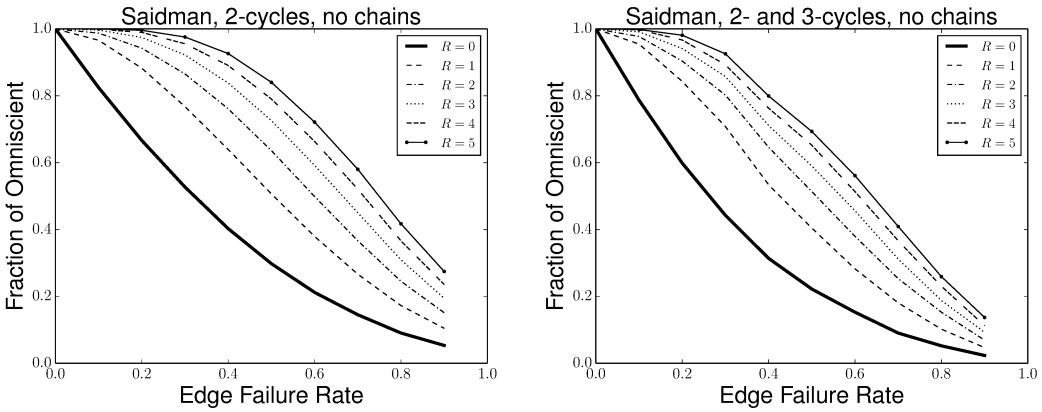

Fig. 2. Saidman generator graphs constrained to 2-cycles only (left) and both 2- and 3-cycles (right).

The utility of even a small number of edge queries is evident in Figure 2. Just a single round of testing ($R = 1$) results in 50.6% of omniscient—compared to just 29.8% with no edge testing—for edge failure probability $f = 0.5$ in the 2-cycle case, and there are similar gains in the 2- and 3-cycle case. For the same failure rate, setting $R = 5$ captures 84.0% of the omnsicient 2-cycle matching and 69.3% in the 2- and 3-cycle case—compared to just 22.2% when no edges are queried. Interestingly, we found no statistical difference between non-adaptive and adaptive matching on these graphs.

7.2. 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 143 transplant centers—that is, 60% of the centers in the U.S.—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 [Ashlagi and Roth 2014; Dickerson et al. 2012b] but does in a sparse model [Ashlagi et al. 2011]) 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 3 shows the fraction of the omniscient 2-cycle and 2-cycle with chains match size achieved by using only 2-cycles or both 2-cycles and chains and some small number of non-adaptive edge query rounds $R \in \{0, 1, \ldots, 5\}$. For each of the 169 pre-test compatibility graphs and each of 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 3 fails, then we include all successful edges (in this case, 2 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 7.1. Still, the gain realized even by a small number of edge query rounds is stark, with $R = 5$ achieving over 90% of the omniscient objective for every failure rate

in the 2-cycles-only case, and over 75% of the omniscient objective when chains are included (and typically much more).

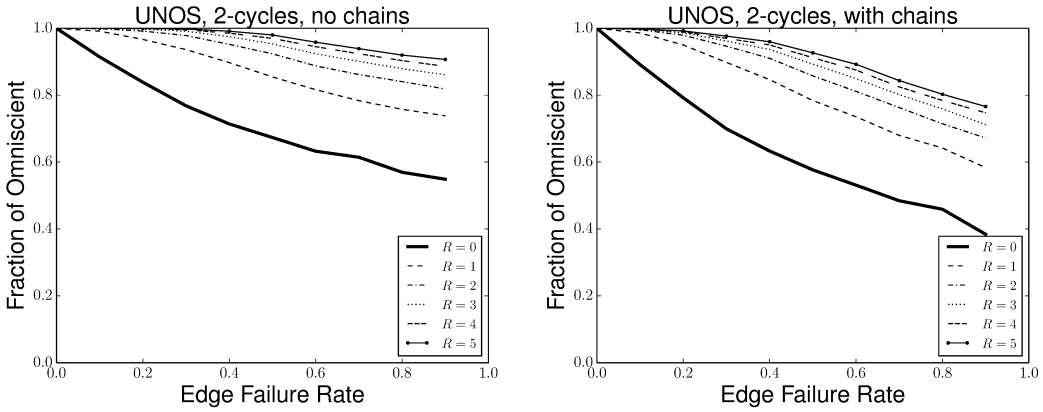

Fig. 3. Real UNOS match runs constrained to 2-cycles (left) and both 2-cycles and chains (right).

Figure 4 expands these results to the case with 2- and 3-cycles, both without and with chains. Slightly less of the omniscient matching objective is achieved across the board, but the overall increases due to $R \in \{1, \ldots, 5\}$ non-adaptive rounds of testing is once again prominent. Interestingly, we did not see a significant difference in results for adaptive and non-adaptive edge testing on the UNOS family of graphs, either.

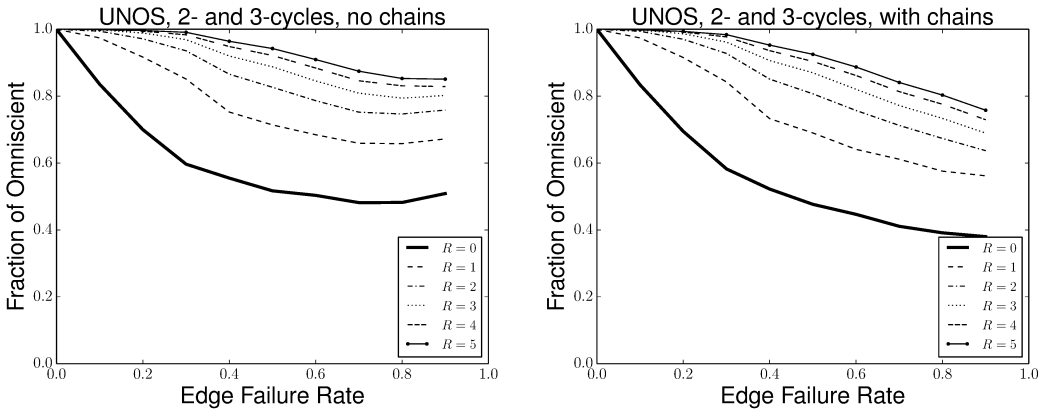

Fig. 4. Real UNOS match runs with 2- and 3-cycles and no chains (left) and with chains (right).

We provide additional experimental results in Appendix D. Code to replicate all experiments is available at https://github.com/JohnDickerson/KidneyExchange; this codebase includes graph generators but, due to privacy concerns, does not include the real match runs from the UNOS exchange.

8. CONCLUSIONS & FUTURE RESEARCH

In this paper, we addressed stochastic matching and its generalization to k-set 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 *non-adaptive* 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-set packing* by designing an adaptive algorithm that achieves a $(\frac{2}{k}-\epsilon)$ fraction of 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 standard *non-stochastic* setting [Fürer and Yu 2013].

We adapted these algorithms to the kidney exchange problem and, on both generated and real data from the first 169 runs of the UNOS US 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 non-adaptive edge queries per donor-patient pair results in large gains in expected successful matches across a wide range of edge failure probabilities.

8.1. Open theoretical problems

Two main open theoretical problems remain open. 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 *non-adaptive 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 k-set packing, we achieve a $(\frac{2}{k}-\epsilon)$-approximation using $O(n)$ queries—in polynomial time. In kidney exchange, however, our scarcest resource is crossmatch tests; computational hardness is circumvented daily, through integer programming techniques [Abraham et al. 2007]. Is there an exponential-time adaptive algorithm for k-set packing that requires $O(1)$ rounds and $O(n)$ queries, and achieves a $(1-\epsilon)$-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-set packing [Hurkens and Schrijver 1989].

8.2. Discussion of policy implications of 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.

More accurate models 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 policymakers who are not necessarily versed in optimization or theoretical economics and computer science—and the organizations they represent rightfully 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. 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 7.2 shows, even a *single* extra edge test per pair will produce substantially better results. Furthermore, these extra edge tests can be performed entirely in parallel if an exchange decides on

a non-adaptive testing strategy (which we showed is quite effective), so the temporal cost to waiting time between a match run and transplant event would be minimal compared to the status quo.

Clearly, more extensive studies would need to be undertaken before an exact policy recommendation could be made. These studies could take factors like the monetary cost of an extra crossmatch test or variability in testing prowess across different medical laboratories into account explicitly during the optimization process. Furthermore, various prioritization schemes could 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. 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.

REFERENCES

ABRAHAM, D. J., BLUM, A., AND SANDHOLM, T. 2007. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC)*. 295–304.

ADAMCZYK, M. 2011. Improved analysis of the greedy algorithm for stochastic matching. *Information Processing Letters 111,* 15, 731–737.

AKBARPOUR, M., LI, S., AND GHARAN, S. O. 2014. Dynamic matching market design. In *Proceedings of the ACM Conference on Economics and Computation (EC)*. 355.

ANDERSON, R., ASHLAGI, I., GAMARNIK, D., AND KANORIA, Y. 2015a. A dynamic model of barter exchange. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1925–1933.

ANDERSON, R., ASHLAGI, I., GAMARNIK, D., AND ROTH, A. E. 2015b. Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences 112,* 3, 663–668.

ASADPOUR, A., NAZERZADEH, H., AND SABERI, A. 2008. Stochastic submodular maximization. In *Proceedings of the 4th International Workshop on Internet and Network Economics (WINE)*. 477–489.

ASHLAGI, I., GAMARNIK, D., REES, M. A., AND ROTH, A. E. 2011. The need for (long) chains in kidney exchange. Manuscript.

ASHLAGI, I., JAILLET, P., AND MANSHADI, V. H. 2013. Kidney exchange in dynamic sparse heterogenous pools. In *Proceedings of the 14th ACM Conference on Electronic Commerce (EC)*. 25–26.

ASHLAGI, I. AND ROTH, A. 2014. Free riding and participation in large scale, multi-hospital kidney exchange. *Theoretical Economics*. Forthcoming; preliminary version in EC'11.

AWASTHI, P. AND SANDHOLM, T. 2009. Online stochastic optimization in the large: Application to kidney exchange. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*. 405–411.

BANSAL, N., GUPTA, A., LI, J., MESTRE, J., NAGARAJAN, V., AND 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, A. D., AND SHARMA, A. 2013. Harnessing the power of two crossmatches. In *Proceedings of the 14th ACM Conference on Electronic Commerce (EC)*. 123–140.

BOLLOBÁS, B. 2001. *Random Graphs* 2nd Ed. Cambridge University Press.

CHEN, N., IMMORLICA, N., KARLIN, A. R., MAHDIAN, M., AND RUDRA, A. 2009. Approximating matches made in heaven. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP)*. 266–278.

CONSTANTINO, M., KLIMENTOVA, X., VIANA, A., AND RAIS, A. 2013. New insights on integer-programming models for the kidney exchange problem. *European Journal of Operational Research 231,* 1, 57–68.

COSTELLO, K. P., TETALI, P., AND TRIPATHI, P. 2012. Matching with commitment. In *Proceedings of the 39th International Colloquium on Automata, Languages and Programming (ICALP)*. 822–833.

DEAN, B. C., GOEMANS, M. X., AND VONDRAK, J. 2004. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *Proceedings of the 45th Symposium on Foundations of Computer Science (FOCS)*. 208–217.

DICKERSON, J. P., PROCACCIA, A. D., AND SANDHOLM, T. 2012a. Dynamic matching via weighted myopia with application to kidney exchange. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*. 1340–1346.

DICKERSON, J. P., PROCACCIA, A. D., AND SANDHOLM, T. 2012b. Optimizing kidney exchange with transplant chains: Theory and reality. In *Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 711–718.

DICKERSON, J. P., PROCACCIA, A. D., AND SANDHOLM, T. 2013. Failure-aware kidney exchange. In *Proceedings of the 14th ACM Conference on Electronic Commerce (EC)*. 323–340.

DICKERSON, J. P. AND SANDHOLM, T. 2015. FutureMatch: Combining human value judgments and machine learning to match in dynamic environments. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*.

FÜRER, M. AND YU, H. 2013. Approximate the k-set packing problem by local improvements. *CoRR abs/1307.2262*.

GLORIE, K. M., VAN DE KLUNDERT, J. J., AND WAGELMANS, A. P. M. 2014. Kidney exchange with long chains: An efficient pricing algorithm for clearing barter exchanges with branch-and-price. *Manufacturing & Service Operations Management 16,* 4, 498–512.

GOEL, G. AND TRIPATHI, P. 2012. Matching with our eyes closed. In *Proceedings of the 53rd Symposium on Foundations of Computer Science (FOCS)*. 718–727.

GUPTA, A., KRISHNASWAMY, R., NAGARAJAN, V., AND RAVI, R. 2012. Approximation algorithms for stochastic orienteering. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1522–1538.

GUPTA, A. AND NAGARAJAN, V. 2013. A stochastic probing problem with applications. In *Proceedings of the 16th Conference on Integer Programming and Combinatorial Optimization (IPCO)*. 205–216.

HURKENS, C. A. J. AND 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 Journal on Discrete Mathematics 2,* 1, 68–72.

LEISHMAN, R., FORMICA, R., ANDREONI, K., FRIEDEWALD, J., SLEEMAN, E., MONSTELLO, C., STEWART, D., AND SANDHOLM, T. 2013. The Organ Procurement and Transplantation Network (OPTN) Kidney Paired Donation Pilot Program (KPDPP): Review of current results. In *American Transplant Congress (ATC)*. Talk abstract.

MANLOVE, D. AND O'MALLEY, G. 2015. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *ACM Journal of Experimental Algorithmics*. To appear.

ROTH, A. E., SÖNMEZ, T., AND ÜNVER, M. U. 2004. Kidney exchange. *Quarterly Journal of Economics 119,* 2, 457–488.

ROTH, A. E., SÖNMEZ, T., AND ÜNVER, M. U. 2005. Pairwise kidney exchange. *Journal of Economic Theory 125*, 151–188.

ROTH, A. E., SÖNMEZ, T., AND ÜNVER, M. U. 2007. Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review 97,* 3, 828–851.

SAIDMAN, S. L., ROTH, A. E., SÖNMEZ, T., ÜNVER, M. U., AND DELMONICO, F. L. 2006. Increasing the opportunity of live kidney donation by matching for two and three way exchanges. *Transplantation 81*, 773–782.

ÜNVER, M. U. 2010. Dynamic kidney exchange. *Review of Economic Studies 77,* 1, 372–414.

Online Appendix to: Ignorance is Almost Bliss: Near-Optimal Stochastic Matching With Few Queries

AVRIM BLUM, Carnegie Mellon University
JOHN P. DICKERSON, Carnegie Mellon University
NIKA HAGHTALAB, Carnegie Mellon University
ARIEL D. PROCACCIA, Carnegie Mellon University
TUOMAS SANDHOLM, Carnegie Mellon University
ANKIT SHARMA, Solvvy

A. MISSING PROOFS FROM SECTION 4

A.1. Analysis of the Non-Adaptive Algorithm

LEMMA A.1. *Let E_1 be an arbitrary subset of edges of E, and let $E_2 = E \setminus E_1$. Then $\overline{M}(E) \leq \overline{M}(E_1) + \overline{M}(E_2)$.*

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 $|M(E')| \leq |M(E'_1)| + |M(E'_2)|$. 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 $|M(E'_1)| \geq |T \cap E'_1|$ and $|M(E'_2)| \geq |T \cap E'_2|$, and hence $|M(E')| \leq |M(E'_1)| + |M(E'_2)|$. Expectation is a convex combination of the values of the outcomes. For every subset E' of edges in E, multiplying the above 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 $\overline{M}(E) \leq \overline{M}(E_1) + \overline{M}(E_2)$. □

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

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

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

PROOF. Define $U = E \setminus W_{r-1}$. Assume that $\overline{M}(W_{r-1}) \leq \overline{M}(E)/2$. By Lemma A.1, we know that $\overline{M}(U) \geq \overline{M}(E) - \overline{M}(W_{r-1})$. Hence, $|O_r| = |M(U)| \geq \overline{M}(U) \geq \overline{M}(E) - \overline{M}(W_{r-1}) \geq \overline{M}(E)/2$.

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 4.2, 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,

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

DOI: http://dx.doi.org/10.1145/0000000.0000000

where the expectation on the left hand side is taken only over the outcome of the edges in O_r. Therefore, we have $\overline{M}(O_r \cup W'_{r-1}|W'_{r-1}) \geq \frac{\alpha}{2}\,\overline{M}(E) + (1-\gamma)|M(W'_{r-1})|$, 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, i.e.,

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

□

PROOF OF THEOREM 5.1. For ease of exposition, assume $L = \frac{4}{\epsilon} - 1$ is an odd integer. Then, either $\overline{M}(W_R) \geq \overline{M}(E)/2$ in which case we are done. Or otherwise, by repeatedly applying Lemma A.2 for R steps, we have

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

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. □

A.2. Example Graph for the Non-Adaptive Algorithm

LEMMA A.3. *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 will complete our claim.

In order 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 will follow.

Consider any set $S \subseteq U$ of size at most $n - c\log(n)$. We will 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. □

PROOF OF THEOREM 5.2. Let (V, E) be a graph, illustrated in Figure 5, 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, $\overline{M}(E)$. Since $p = 0.5$, in expectation, half of the edges in the perfect matching between B and C exist, and therefore half of the vertices of B

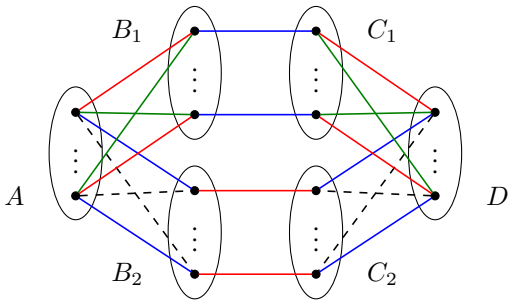

Fig. 5. The blue and red edges represent the matching picked at rounds 1 and 2, respectively. The green edges represent the edges picked at round 3 and above. The dashed edges are never picked by the algorithm.

and C will get matched. By Lemma A.3, 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 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 half of the vertices in B. Similarly, define C_1 and C_2. Since 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 $\frac{5}{4}t$ $(= \frac{5}{6}\ \frac{3}{2}t)$.

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 algorithms 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 $\frac{1}{4}(\frac{t}{2} + \frac{t}{2}) = \frac{t}{4}$. Since the vertices in A can only be matched with vertices in B, and the vertices in D can only be matched with vertices in C, it follows that at least $\frac{t}{4}$ of the vertices in A and C are unmatched in expectation. Hence, the total number of edges included in the matching is at most $\frac{5}{4}t$. This completes our claim. □

B. MISSING PROOFS FROM SECTION 5

In this section, we fill in the missing proofs for stochastic k-set packing. A notation that we will use in some parts of the analysis is $\overline{K}(A|B)$ that we define as follows: Given a collection $B \subseteq A$ that has been queried and $B' \subseteq B$ that exists, we use $\overline{K}(A|B)$ to denote $\mathbb{E}[|K(X_p \cup B')|]$ where X_p is the random set formed by including every element of $A \setminus B$ independently with probability p.

B.1. Adaptive Algorithm for k-Set Packing

We introduce some notation that is used in the remainder of the proofs in this section. At the beginning of the r^{th} iteration of Algorithm 4, we know the results of the queries $\bigcup_{i=1}^{r-1} Q_i$. We define Z_r to be the expected cardinality of the instance (U, A) given the result of these queries. More formally, $Z_r = \overline{K}(A|\bigcup_{i=1}^{r-1} Q_i)$. We note that $Z_1 = \overline{K}(A)$.

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 over *only* 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 $\mathbb{E}_{Q_r,\ldots,Q_R}[X]$ 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 $\bigcup_{i=1}^{r-1} Q_i$.

The next result, Lemma B.1, proves a lower bound on the expected increase in the cardinality of B_r (the solution at round r) with respect to B_{r-1} (the solution in the previous round).

LEMMA B.1. *For every $r \in [R]$, it is the case that $\mathbb{E}_{Q_r}[|B_r|] \geq (1-\gamma)|B_{r-1}| + \gamma(\frac{2}{k} - \eta)Z_r$, where $\gamma = \frac{p^{s_{\eta,k}}}{(\frac{2}{k}-\eta)k\, s_{\eta,k}}$.*

PROOF. By Lemma 6.2, Q_r is a collection of at least $\frac{1}{k\, s_{\eta,k}}(|K(A_r)| - \frac{|B_{r-1}|}{\frac{2}{k}-\eta})$ *disjoint $s_{\eta,k}$-size augmenting structures* (C, D) for B_{r-1}. Since in each augmenting structure (C, D), C has at most $s_{\eta,k}$ sets, on querying, the set C exists with probability at least $p^{s_{\eta,k}}$. Therefore, the expected increase in the size of the solution at Step 2c is:

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

Noting that $|K(A_r)| \geq Z_r$, we have our result. □

PROOF OF THEOREM 6.3. First, we make a technical observation about Z_r: For every $r \leq R$, $\mathbb{E}_{Q_{r-1}}[Z_r] = Z_{r-1}$. This is since

$$\mathbb{E}_{Q_{r-1}}[Z_r] = \mathbb{E}_{Q_{r-1}}[\overline{K}(A| \bigcup_{i=1}^{r-1} Q_i)] = \overline{K}(A| \bigcup_{i=1}^{r-2} Q_i) = Z_{r-1}. \tag{4}$$

Now, similar to the proof of Theorem 4.1, we first apply Lemma B.1 to the R^{th} step and get $\mathbb{E}_{Q_R}[|B_R|] \geq (1-\gamma)|B_{R-1}| + \gamma(\frac{2}{k} - \eta)Z_R$. Next taking expectation on both sides with respect to Q_{R-1}, we get $\mathbb{E}_{Q_{R-1},Q_R}[|B_R|] \geq (1-\gamma)\mathbb{E}_{Q_{R-1}}[|B_{R-1}|] + \gamma(\frac{2}{k} - \eta)\mathbb{E}_{Q_{R-1}}[Z_R]$. Applying Lemma B.1 to $\mathbb{E}_{Q_{R-1}}[|B_{R-1}|]$ and Equation (4) to $\mathbb{E}_{Q_{R-1}}[Z_R]$, we get

$$\begin{aligned}\mathbb{E}_{Q_{R-1},Q_R}[|B_R|] &\geq (1-\gamma)((1-\gamma)|B_{R-2}| + \gamma(\frac{2}{k} - \eta)Z_{R-1}) + \gamma(\frac{2}{k} - \eta)\, Z_{R-1} \\ &= (1-\gamma)^2|B_{R-2}| + \gamma(\frac{2}{k} - \eta)(1 + (1-\gamma))\, Z_{R-1}.\end{aligned}$$

We can repeat the above steps, by sequentially taking expectation over Q_{R-2} through Q_1, and applying Lemma B.1 and Equation (4) at each step, to achieve

$$\begin{aligned}\mathbb{E}_{Q_1,\ldots,Q_R}[|B_R|] &\geq (1-\gamma)^R|B_0| + \gamma(\frac{2}{k} - \eta)(1 + (1-\gamma) + \cdots + (1-\gamma)^{R-1})\, Z_1 \\ &\geq (\frac{2}{k} - \eta)(1 - (1-\gamma)^R)\, \overline{K}(A) \geq \frac{2}{k}(1 - \frac{\eta k}{2})(1 - e^{-\gamma R}).\, \overline{K}(A)\end{aligned}$$

We complete the claim by noting that

$$\frac{2}{k}(1 - \frac{\eta k}{2})(1 - e^{-\gamma R}) \geq \frac{2}{k}(1 - \frac{\epsilon}{2})(1 - \frac{\epsilon}{2}) \geq (1-\epsilon)\frac{2}{k},$$

where the penultimate inequality comes from the fact that $\eta = \epsilon/k$ and

$$R = \frac{(\frac{2}{k} - \eta)\, k\, s_{\eta,k}}{p^{s_{\eta,k}}} \log(\frac{2}{\epsilon}) = \frac{1}{\gamma} \log(\frac{2}{\epsilon}).$$

Therefore, the cardinality of B_R in expectation is at least a $(1-\epsilon)\frac{2}{k}\overline{K}(A)$. □

B.2. Non-Adaptive Algorithm for k-Set Packing

To prove Theorem 6.4, we analyze the non-adaptive Algorithm 5 from the main paper. Before proving Theorem 6.4, we present a technical claim.

CLAIM B.2. *Let $A_1 \subseteq A$ and $A_2 = A \setminus A_1$. Then $\overline{K}(A) \leq \overline{K}(A_1) + \overline{K}(A_2)$.*

PROOF. Let A' be any subset of A, $A'_1 = A_1 \cap A'$, and $A'_2 = A_2 \cap A'$. Since the k-set packing of A' restricted to A'_1 and A'_2 are valid k-set packings for these subsets, hence $|K(A')| \leq |K(A'_1))| + |K(A'_2)|$. For every $A' \subseteq A$, the above inequality holds. Expectation is a linear combination of the values of the outcomes, and so this inequality also holds in expectation. That is, $\overline{K}(A) \leq \overline{K}(A_1) + \overline{K}(A_2)$. $\square$

PROOF OF THEOREM 6.4. We claim that the expected cardinality of the k-set solution output by Algorithm 5 is at least $(1 - \frac{\epsilon}{2})\frac{(\frac{2}{k}-\eta)^2}{1+\frac{2}{k}-\eta}\overline{K}(A)$. The claimed approximation will follow since $\eta = \frac{\epsilon}{2k}$.

For ease of exposition, let $\alpha = \frac{\frac{2}{k}-\eta}{1+\frac{2}{k}-\eta}$, and now note that $\frac{(\frac{2}{k}-\eta)^2}{1+\frac{2}{k}-\eta} = \alpha(\frac{2}{k}-\eta) = (1-\alpha)(\frac{2}{k}-\eta)^2$.

Assume that $\overline{K}(B_R) \leq \alpha \cdot \overline{K}(A)$ (else it will be immediately follow that the expected cardinality of the k-set solution output by the algorithm is at least $(\frac{2}{k} - \eta)\alpha\overline{K}(A)$ and this will complete the claim).

First, we make an observation. For each round $r \in [R]$, we have $\overline{K}(B_r) \leq \overline{K}(B_R) \leq \alpha\overline{K}(A)$. If we denote $A_r = A \setminus B_{r-1}$, then it follows that

$$|O_r| \geq (\frac{2}{k}-\eta)|K(A_r)| \geq (\frac{2}{k}-\eta)\overline{K}(A_r) \geq (\frac{2}{k}-\eta)(\overline{K}(A)-\overline{K}(B_{r-1})) \geq (\frac{2}{k}-\eta)(1-\alpha)\overline{K}(A)\,,$$

where the first inequality follows from the fact that O_r is $(\frac{2}{k} - \eta)$-approximation to A_r, and the second inequality follows from Claim B.2.

We analyze the expected cardinality of the output solution Q_R by analyzing the R stages that the algorithm adopts at Steps 3 and 4 to create solution Q_R. For this analysis, we use the following notation: For a given r, we use the notation $\mathbb{E}_{O_r}[X]$ to denote the expected value of X where the expectation is taken over *only* the outcome of query O_r, and fixing the outcomes on the results of queries $\bigcup_{i=1}^{r-1} O_i$. Moreover, for a given r, we use $\mathbb{E}_{O_r,\ldots,O_R}[X]$ to denote the expected value of X with the expectation taken over the outcomes of queries $\bigcup_{i=r}^{R} O_i$, and fixing an outcome on the results of queries $\bigcup_{i=1}^{r-1} O_i$.

In the first stage, Q_1 is assigned to the collection of k-sets that are found to exist in O_1. In the second stage, we try to augment Q_1 by finding augmenting structures from O_2 and querying them. By Lemma 6.2, it finds at least $\frac{1}{ks_{\eta,k}}\left(|O_2| - \frac{|Q_1|}{\frac{2}{k}-\eta}\right)$ disjoint augmenting structures from O_2 that have size at most $s_{\eta,k}$ and augment Q_1. Since each augmenting structure exists independently with probability at least $p^{s_{\eta,k}}$, *in expectation over the outcomes of queries to* O_2, the size of Q_2, $\mathbb{E}_{O_2}[Q_2]$, is at least

$$\begin{aligned}|Q_1| + p^{s_{\eta,k}}\left(\frac{1}{ks_{\eta,k}}(|O_2| - \frac{|Q_1|}{\frac{2}{k}-\eta})\right) =& \frac{p^{s_{\eta,k}}}{ks_{\eta,k}}\,|O_2| + (1 - \frac{p^{s_{\eta,k}}}{ks_{\eta,k}(\frac{2}{k}-\eta)})|Q_1| \\ \geq& \frac{p^{s_{\eta,k}}}{ks_{\eta,k}}\,(\frac{2}{k}-\eta)(1-\alpha)\overline{K}(A) + (1 - \frac{p^{s_{\eta,k}}}{ks_{\eta,k}(\frac{2}{k}-\eta)})\,|Q_1|,\end{aligned}$$

and hence the expected size of Q_2 is at least $\beta\,\overline{K}(A) + (1-\gamma)|Q_1|$, where $\beta = \frac{p^{s_{\eta,k}}}{ks_{\eta,k}}(\frac{2}{k}-\eta)(1-\alpha)$ and $\gamma = \frac{p^{s_{\eta,k}}}{k\ s_{\eta,k}\ (\frac{2}{k}-\eta)}$.

For the third stage, a similar analysis shows that the expected size of Q_3, $\mathbb{E}_{O_3}[Q_3]$, *with expectation taken only over the outcomes of the queries to* O_3, is at least $\beta\ \overline{K}(A) + (1-\gamma)|Q_2|$.

If we now, in addition, take expectation over the outcomes of queries to O_2, we get the expected size of Q_3, $\mathbb{E}_{O_2,O_3}[Q_3]$, is at least $\beta\ \overline{K}(A) + (1-\gamma)\ (\beta\ \overline{K}(A) + (1-\gamma)|Q_1|) = \beta(1+(1-\gamma))\ \overline{K}(A) + (1-\gamma)^2\ |Q_1|$.

Repeating the above steps, the procedure creates the k-set solution Q_R (from $O_1, \cdots, O_R$) whose expected size, with expectation taken over the outcomes of queries to O_2 through O_R, is at least

$$\beta(1+(1-\gamma)+\cdots+(1-\gamma)^{R-2})\overline{K}(A) + (1-\gamma)^{R-1}|Q_1|\ .$$

Finally, taking expectation over outcomes of queries to O_1, since the expected size of $|Q_1|$ is at least $p|O_1| \geq p\ (\frac{2}{k}-\eta)\overline{K}(A) \geq \beta\ \overline{K}(A)$, we have that the expected size of Q_R is at least

$$\begin{aligned}&\beta\ (1+(1-\gamma)+\cdots+(1-\gamma)^{R-1})\overline{K}(A)\\ &\quad= \frac{\beta}{\gamma}(1-(1-\gamma)^R)\overline{K}(A) \geq \frac{\beta}{\gamma}(1-e^{-\gamma R})\overline{K}(A) \geq (1-\frac{\epsilon}{2})\frac{(\frac{2}{k}-\eta)^2}{\frac{2}{k}-\eta+1}\overline{K}(A)\end{aligned}$$

□

C. MATCHING UNDER CORRELATED EDGE PROBABILITIES

In this section, we extend our framework to a more general setting. Here, the existence probability of an edge depends on parameters that are associated with the endpoints of the edge. Specifically, every vertex $v_i \in V$ is associated with parameter p_i, and an edge $e_{ij} = (v_i, v_j)$ exists with probability $p_i p_j$.

Importantly, this model is a generalization of the model studied above: we can still think of each edge $e \in E$ as existing with a given probability, and these events are *independent*. However, using vertex parameters gives us a formal framework for correlating the probabilities of edges incident to any particular vertex. The motivation for this comes from kidney exchange: Some *highly sensitized* patients are less likely than other patients to be compatible with potential donors. Such patients correspond to a small p_i parameter.

We consider two settings: adversarial and stochastic. In the adversarial setting, the vertex parameters p_i are selected by an adversary, whereas in the stochastic model, the parameters are drawn from a distribution. In the former setting, for $\delta > 0$, define f_δ to be the *number of vertices* that have $p_i < \delta$. In the latter setting, for a distribution D and $\delta > 0$, let g_δ indicate the *probability* that a vertex has its parameter less than δ, i.e., $g_\delta = \Pr_{p_i \sim D}[p_i < \delta]$. We formulate our results in terms of δ, f_δ, and g_δ, and the desired value of δ can depend on the application. For example, in kidney exchange, δ would be the probability that a highly-sensitized patient is compatible with a random donor (a patient is typically considered to be highly sensitized when this probability is 0.2), and f_δ would be the number of highly-sensitized patients in the kidney exchange pool.

C.1. Adaptive Algorithm in Adversarial Setting

In this section, we consider the case where an adversary chooses the values of vertex parameters. We give guarantees on the performance of Algorithm 1 in this setting.

THEOREM C.1. *For any graph (V, E), any $\epsilon > 0$, and $\delta > 0$, Algorithm 1 returns a matching with expected size of $(1-\epsilon)(\overline{M}(E) - f_\delta)$ in $R = \frac{\log(2/\epsilon)}{\delta^{4/\epsilon}}$ iterations.*

The proof of this theorem and the subsequent lemmas are similar to the proofs of Section 4, and are included here for completeness. In the next lemma, $\mathbb{E}_{Q_r}[|M_r|]$ indicates the expected size of M_r, where the expectation is over the query outcome of Q_r. More formally, $\mathbb{E}_{Q_r}[|M_r|] = \overline{M}(\bigcup_{j=1}^{r} Q_j | \bigcup_{j=1}^{r-1} Q_j)$. We use Z_r to denote the expected size of the maximum matching in graph (V, E) given the results of the queries $\bigcup_{j=1}^{r-1} Q_j$. More formally, $Z_r = \overline{M}(E | \bigcup_{j=1}^{r-1} Q_i)$. Note that $Z_1 = \overline{M}(E)$.

LEMMA C.2. *For all $r \in [R]$ and odd L, $\mathbb{E}_{Q_r}[|M_r|] \geq (1-\gamma)|M_{r-1}| + \alpha(Z_r - f_\delta)$, where $\gamma = \delta^{L+1}(1+\frac{2}{L+1})$ and $\alpha = \delta^{L+1}$.*

PROOF. By Lemma 4.2, there exists $|O_r| - (1+\frac{2}{L+1})|M_{r-1}|$ many augmenting paths in $O_r \Delta M_{r-1}$ that augment M_{r-1} and have length at most L. These augmenting paths are disjoint, so at most f_δ of them include a vertex v_i, with $p_i \leq \delta$. We will ignore these paths. Among the remaining augmenting paths, each path of length L, has at most $\frac{L+1}{2}$ edges that have not been queried yet. These edges do not share a vertex, so each one exists, independently of others, with probability at least δ^2. Therefore, the expected increase in the size of the matching from these augmenting paths is:

$$\mathbb{E}_{Q_r}[|M_r|] - |M_{r-1}| \geq \delta^{L+1}\left(|O_r| - (1+\frac{2}{L+1})|M_{r-1}| - f_\delta\right) \geq \alpha(Z_r - f_\delta) - \gamma|M_{r-1}|.$$

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

PROOF SKETCH OF THEOREM C.1. Let $L = \frac{4}{\epsilon} - 1$. First note that for all r, it is true that

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

The remainder of the proof is similar to that of Theorem 4.1 with $Z_r - f_\delta$ replacing Z_r. Following similar analysis, we have

$$\mathbb{E}_{Q_1,\ldots,Q_R}[|M_R|] \geq \alpha\frac{1-(1-\gamma)^R}{\gamma}(\overline{M}(E) - f_\delta).$$

Since $R = \frac{\log(2/\epsilon)}{\delta^{4/\epsilon}}$, we have

$$\frac{\alpha}{\gamma}\big(1-(1-\gamma)^R\big) \geq (1-\frac{2}{L+1})\big(1-(1-\gamma)^R\big) \geq (1-\frac{\epsilon}{2})(1-e^{-\gamma R}) \geq (1-\epsilon). \quad (5)$$

Therefore, Algorithm 1 returns a matching with expected size of $(1-\epsilon)(\overline{M}(E) - f_\delta)$. □

C.2. Adaptive Algorithm in Stochastic Setting

In this section, we consider the case where the vertex parameters are drawn independently from a distribution.

COROLLARY C.3. *Given any graph (V, E) with vertex parameters that are drawn from distribution D and any $\epsilon, \delta > 0$, Algorithm 1 returns a matching with expected size of $(1-\epsilon)(\overline{M}(E) - ng_\delta)$ in $R = \frac{\log(2/\epsilon)}{\delta^{4/\epsilon}}$ iterations.*

PROOF. The result of Theorem C.1 holds for any value of f_δ. Hence, on taking expectation over the value of f_δ, we have our result. □

The next corollary shows the implication of Corollary C.3 for the uniform distribution.

COROLLARY C.4. *For a given graph (V, E) with vertex parameters that are drawn from the uniform distribution, and any $\epsilon > 0$, Algorithm 1 returns a matching with expected size of $(1-\epsilon)(\overline{M}(E) - \epsilon n)$ in $R = \frac{\log(2/\epsilon)}{\epsilon^{4/\epsilon}}$ iterations.*

PROOF. This follows from Corollary C.3 by setting $\delta = \epsilon$ and noting that $g_\epsilon = \epsilon$ for the uniform distribution. □

C.3. Non-adaptive algorithm in Adversarial Setting

In this section, we consider the case where an adversary chooses the values of vertex parameters. We prove performance guarantees for Algorithm 2 in this adversarial setting.

THEOREM C.5. *Given a graph (V, E) with vertex parameters that are selected by an adversary, and any $\epsilon, \delta > 0$, Algorithm 2 returns a matching with expected size of $\frac{1}{2}(1-\epsilon)(\overline{M}(E) - f_\delta)$ in $R = \frac{\log(2/\epsilon)}{\delta^{4/\epsilon}}$ iterations.*

The proof of Theorem C.5 and the subsequent lemma are similar to Section 5, and are included here for completeness.

LEMMA C.6. *For any iteration $r \in [R]$ of Algorithm 2 and odd L, if $\overline{M}(W_{r-1}) \leq \overline{M}(E)/2$, then $\overline{M}(W_r) \geq \frac{\alpha}{2}\,(\overline{M}(E) - f_\delta) + (1-\gamma)\overline{M}(W_{r-1})$, where $\alpha = \delta^{L+1}$ and $\gamma = \delta^{L+1}(1 + \frac{2}{L+1})$.*

PROOF. Define $U = E \setminus W_{r-1}$. Assume that $\overline{M}(W_{r-1}) \leq \overline{M}(E)/2$. By Claim A.1, we know that $\overline{M}(U) \geq \overline{M}(E) - \overline{M}(W_{r-1})$. Hence, $|O_r| = |M(U)| \geq \overline{M}(U) \geq \overline{M}(E) - \overline{M}(W_{r-1}) \geq \overline{M}(E)/2$.

Let W'_{r-1} represent one possible outcome of existing edges when edges are drawn from W_{r-1}. By Lemma 4.2, 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 $M(W'_{r-1})$. Among these paths, at most f_δ have a vertex v_i, with $p_i < \delta$. We ignore these paths. Each remaining path succeeds with probability $(\delta^2)^{(L+1)/2}$. Hence, the expected increase in the size of $|M(W'_{r-1})|$ using the remaining paths of length L is,

$$\begin{aligned}\overline{M}(O_r \cup W'_{r-1}|W'_{r-1}) - |M(W'_{r-1})| &\geq \delta^{L+1}\left(|O_r| - (1+\frac{2}{L+1})|M(W'_{r-1})| - f_\delta\right)\\ &\geq \delta^{L+1}\left(\frac{1}{2}\,\overline{M}(E) - (1+\frac{2}{L+1})|M(W'_{r-1})| - f_\delta\right).\end{aligned}$$

Re-arranging the inequality, we get $\overline{M}(O_r \cup W'_{r-1}|W'_{r-1}) \geq \frac{\alpha}{2}\,(\overline{M}(E) - f_\delta) + (1-\gamma)|M(W'_{r-1})|$. Taking expectation over the coin tosses on W_{r-1} that create outcome W'_{r-1}, we have

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

□

PROOF SKETCH OF THEOREM C.5. Let $L = \frac{4}{\epsilon} - 1$. The proof is similar to that of Theorem 5.1 with the value of $\overline{M}(E)$ being replaced by $\overline{M}(E) - f_\delta$. Following a similar analysis, we get

$$\overline{M}(W_R) \geq \frac{\alpha}{2}\frac{(1-(1-\gamma)^R)}{\gamma}(\overline{M}(E) - f_\delta).$$

Now, $\frac{\alpha}{\gamma}(1-(1-\gamma)^R) \geq (1 - \frac{2}{L+1})(1 - e^{-\gamma R}) \geq (1-\epsilon)$ for $R = \frac{\log(2/\epsilon)}{\delta^{4/\epsilon}}$. Hence, Algorithm 2 returns a matching with expected size of $0.5(1-\epsilon)(\overline{M}(E) - f_\delta)$. □

C.4. Non-adaptive algorithm in Stochastic Setting

We examine the performance of Algorithm 2 in the setting where the vertex parameters are chosen independently from a distribution.

COROLLARY C.7. *Given a graph (V, E) with vertex parameters that are selected from distribution D, and $\epsilon, \delta > 0$, Algorithm 2 returns a matching with expected size of $\frac{1}{2}(1-\epsilon)(\overline{M}(E) - ng_\delta)$ with $R = \frac{\log(2/\epsilon)}{\delta^{4/\epsilon}}$ non-adaptive queries.*

PROOF. The result of Theorem C.5 holds for any value of f_δ. Hence, on taking expectation over the values of f_δ, we have our result. □

COROLLARY C.8. *For any $G = (V, E)$ with vertex parameters that are drawn from the uniform distribution, and any $\epsilon > 0$, Algorithm 2 returns a matching with expected size of $0.5(1-\epsilon)(\overline{M}(E) - n\epsilon)$ with $R = \frac{\log(2/\epsilon)}{\epsilon^{4/\epsilon}}$ non-adaptive queries.*

PROOF. This follows from Corollary C.7 by setting $\delta = \epsilon$ and noting that $g_\epsilon = \epsilon$ for the uniform distribution. □

D. ADDITIONAL EXPERIMENTAL RESULTS ON UNOS COMPATIBILITY GRAPHS

In this section, we include additional experimental results on the same 169 compatibility graphs drawn from the real UNOS kidney exchange used in Section 7. These experiments mimic those of Section 7.2, only this time including in the analysis empty omniscient matchings. If an omniscient matching is empty, then our algorithm will achieve at most zero matches as well. In the body of this paper, we removed these cases from the 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. In this section, we include those cases and rerun the analysis.

Figure 6 mimics Figure 3 from the body of this paper. It shows results for 2-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.

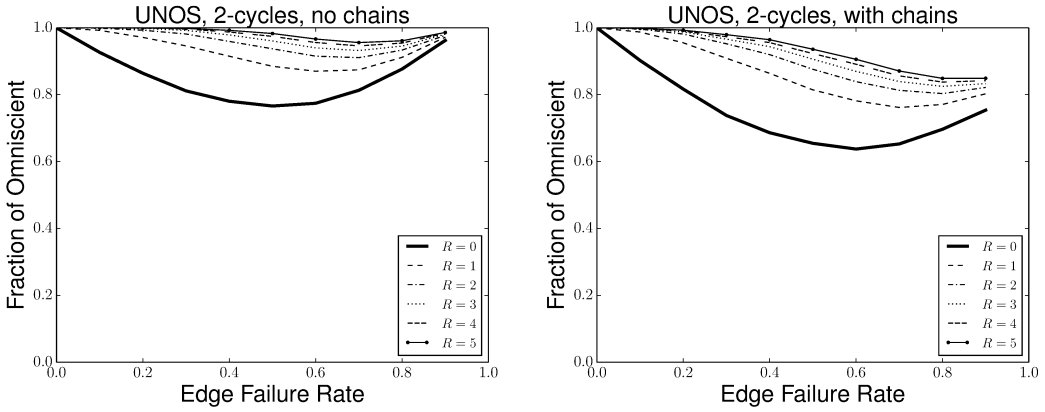

Fig. 6. Real UNOS match runs, restricted matching of 2-cycles only, without chains (left) and with chains (right), including zero-sized omnsicient matchings.

Figure 7 shows the same experiments as Figure 6, only this time allowing both 2- and 3-cycles, without (left) and with (right) chains. It corresponds to Figure 4 in the body of this paper, and exhibits similar but weaker behavior to Figure 6 for high failure rates. This demonstrates the power of including 3-cycles in the matching algorithm—we see that far fewer compatibility graphs admit no matchings under this less-restrictive matching policy.

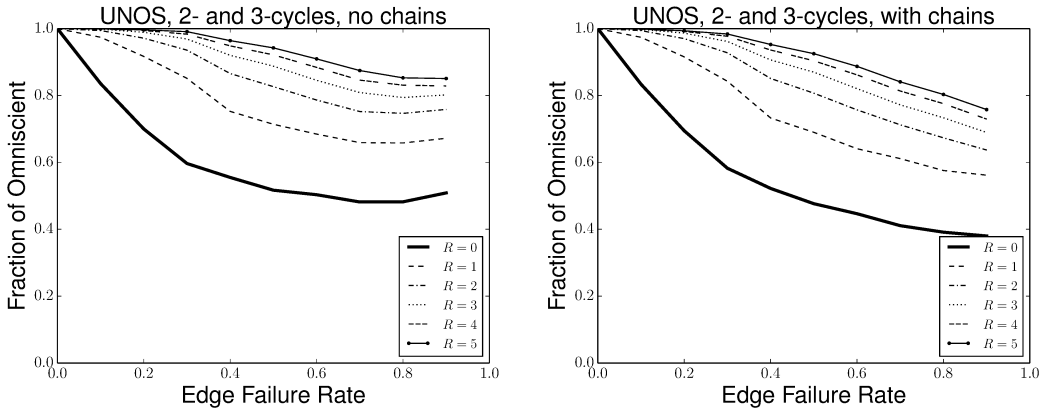

Fig. 7. Real UNOS match runs, matching with 2- and 3-cycles, without chains (left) and with chains (right), including zero-sized omnsicient matchings.