Harnessing the Power of Two Crossmatches

AVRIM BLUM, Carnegie Mellon University
ANUPAM GUPTA, Carnegie Mellon University
ARIEL D. PROCACCIA, Carnegie Mellon University
ANKIT SHARMA, Carnegie Mellon University

Kidney exchanges allow incompatible donor-patient pairs to swap kidneys, but each donation must pass three tests: blood, tissue, and crossmatch. In practice a matching is computed based on the first two tests, and then a single crossmatch test is performed for each matched patient. However, if two crossmatches could be performed per patient, in principle significantly more successful exchanges could take place. In this paper, we ask: If we were allowed to perform two crossmatches per patient, could we harness this additional power optimally and efficiently? Our main result is a polynomial time algorithm for this problem that almost surely computes optimal — up to lower order terms — solutions on random large kidney exchange instances.

1. INTRODUCTION

People who suffer from chronic kidney disease are best treated by transplanting a healthy kidney from a live donor. However, even patients who are fortunate enough to have a willing donor (typically a family member or a close friend) may be incompatible with him. This is where the recent innovation of *kidney exchange* comes in. The basic insight that drives kidney exchange is that two incompatible donor-patient pairs may be able to exchange kidneys so that both patients receive a healthy kidney. To pinpoint as many of these life-saving opportunities as possible, matching algorithms are run (on a weekly or monthly basis) on databases that contain the information of registered of donors and patients.

There are three hurdles that must be cleared before a donation can take place. First, the donor and patient must pass a *blood typing* test. There are four blood types (O, A, B, AB) – depending on the presence of A and B antigens — and only some are compatible with others. For example, a donor with blood type A can donate to a patient with blood type A or AB, but not to a patient with blood type B or O. Second, the donor and patient must pass a *tissue typing* test. There are six tissue antigens; the more of them are shared by the patient and donor, the more likely it is that the transplant will be successful. Third, a *crossmatch* test is performed by (roughly speaking) mixing the donor and patient's blood in a tube and spinning it; depending on whether the blood is suspended or stuck together, doctors can predict whether the patient's body would attack the new kidney (confusingly called *positive crossmatch*) or would accept it (*negative crossmatch*).

The blood and tissue typing tests are fundamentally different from the crossmatch test, in that the relevant information can be collected from each donor and each patient even before matches are made. In contrast, for a crossmatch test (samples of) the blood of the patient and his intended donor must be physically in the same place. Therefore, existing kidney exchanges such as the one run by the United Network for Organ Sharing (UNOS) first compute a matching based only on blood typing and tissue typing tests. Then, crossmatches are performed only for patients and donors that were matched. Exchanges where all the relevant crossmatches are negative proceed to the operating room, while exchanges that involved a positive crossmatch fail.

In graph-theoretic terms, each incompatible donor-patient pair is represented by a vertex. We consider the undirected case where there is an edge between two vertices

Authors' addresses: A. Blum, A. Gupta, A. D. Procaccia, A. Sharma, Computer Science Department, Carnegie Mellon University, email: {avrim,anupamg,arielpro,ankits}@cs.cmu.edu.

if each donor is compatible with the other patient *in terms of blood type and tissue type only*, that is, a pairwise exchange is potentially possible if the crossmatch test is negative[1]. Given a matching on this graph, a crossmatch is performed for each edge in the matching; we model this as flipping an independent coin with some bias p for each crossmatch to determine whether it is positive or negative.

In existing kidney exchanges each donor-patient pair is involved in at most *one* crossmatch test. In this case, to maximize the expected number of transplants we simply need to compute a maximum cardinality matching M on the given graph, since the expected number of transplants is then $p|M|$.

Now imagine a situation where we perform *two* crossmatches per donor-patient pair, instead of one. If for example $p = 0.5$ then, instead of a 50% chance, a patient's odds of receiving a kidney could be as high as 75%. How would we use this additional power to optimize the expected number of transplants? In technical terms, our problem is:

> DOUBLE-CROSSMATCH: *Given a graph $G = (V, E)$, select a subset $E' \subseteq E$ such that for every $v \in V$ there are at most two edges in E' that are incident to v, so that after we throw a coin for each $e \in E'$ to determine whether it exists, the expected size of the maximum cardinality matching on the edges in E' that exist is maximized.*

We aim to construct a polynomial time algorithm that guarantees almost optimal performance with high probability, when the compatibility graph is drawn from a realistic distribution over such graphs. We believe that this practical approach — as opposed to the more standard approach of seeking a constant worst-case multiplicative approximation ratio — can inform policy makers, as we discuss in Section 8.

1.1. Our Results

Before directly tackling realistic kidney exchange models, we investigate our problem on special graphs. In these special cases we can characterize the structure of the optimal solution. While these results are of independent theoretical interest, we also use them as building blocks for our main result.

We first consider the case of a complete (undirected) graph. Note that since we constrain the solution subgraph to have at most two edges incident to a node, hence the edges of the subgraph can be partitioned into cycles and paths. In a complete graph there is no reason to use a path because it is always possible to close it and obtain a cycle. But what is the optimal cycle length? We show that the average gain per vertex is maximized when the selected edges form 4-cycles. In particular, if $|V|$ is divisible by four, then the optimal $|E'|$ consists only of 4-cycles. Moreover, we show that this is true not just for complete graphs. If any graph admits a cover of its vertices through 4-cycles, then *every* optimal solution subgraph would be a 4-cycle cover of the vertices of the graph (Theorem 3.1). For general graphs, this means that our problem is at least as hard as determining whether or not a graph admits a 4-cycle cover, and we leverage this insight to prove the NP-hardness of our problem. Interestingly, closely related problems [Costello et al. 2012; Chen et al. 2009; Bansal et al. 2012] are not known to be — although they are believed to be — NP-hard.

We next analyze the case of complete bipartite graphs. If the two sides of the bipartite graph are not equal in size, then it does not admit a 4-cycle cover. Moreover, paths may actually be useful when they begin and end on the same larger side of the bipartite graph. Would the optimal solution include arbitrarily long paths? We show that, without any loss in the solution quality, we can assume that the optimal solution would only use 4-cycles and paths of length at most 5 (Lemma 4.1).

[1] As we discuss in Section 8, in practice kidney exchanges also use directed 3-cycles.

Using these two results, we move to tackling the case of kidney exchange graphs. As a first step, in Section 6, we consider the case of a complete kidney exchange graph, where every pair of nodes that are blood-type compatible share an edge, that is, we temporarily ignore the tissue typing tests. We give a solution subgraph that is an optimal solution, but for lower order terms, to the complete kidney exchange graph (Theorem 6.4). Then in Section 7, to capture realistic kidney exchanges, we draw the graph G from a distribution over compatibility graphs that was suggested by Ashlagi and Roth [2011]. We now take tissue typing tests into account and pairs of vertices that are blood-type compatible share an edge only if, in addition, they pass a tissue typing test; this occurs with constant probability. Hence, every edge of the complete kidney exchange graph exists with a constant probability in the realistic kidney exchange graph. Our main result (Theorem 7.1) is a polynomial time algorithm with the following property: as the number of vertices goes to infinity, the probability (over the realistic distribution over graphs) that the algorithm fails to select an expectation-maximizing collection of edges, up to lower order terms, goes to zero.

1.2. Related Work

Variants of our problem have been studied under the names *stochastic matching* and the *query-commit problem*. Costello et al. [2012] consider the following version of the problem: Given a random graph G with known edge probabilities p_e, in what order should one query the edges in order to maximize the expected cardinality of the matching? The additional constraint they have is that if on querying the edge it is found to exist, the algorithm is obliged to include it in the matching. It is clear that the greedy approach gives a 0.5-approximation since it finds a maximal matching. In their paper, they present an algorithm that achieves a competitive ratio of 0.575 against an adversary who knows the actual edges that exist in the graph. Furthermore, they show that no algorithm can get a competitive ratio better than 0.896.

Chen et al. [2009] add an additional constraint to the problem and at the same time restrict the strength of the adversary. Just as in the work of Costello et al. [2012], the edges need to be queried in some order, and if a queried edge exists it must be matched. The additional constraint they add is that for every node v we have a known parameter t_v, and the algorithm is not allowed to query more than t_v edges that are incident to node v. They also restrict the strength of the adversary in that now the adversary has precisely as much knowledge of the instance as the algorithm, and in particular, does not know which edges exist in the graph. They showed that the greedy algorithm, which queries the edges in decreasing order of the edge probabilities, gives a 0.25-approximation. Adamczyk [2011] later improved the analysis to show that the greedy algorithm in fact yields a 0.5-approximation.

Bansal et al. [2012] extend the work of Chen et al. [2009] by considering the weighted version of the problem, where in addition to edge probabilities p_e, each edge has a weight w_e and the objective is maximize the expected *weight* of matching (as opposed to cardinality). They give an LP-based solution that achieves a 0.25-approximation for the case of a weighted general graph and a 0.33-approximation for the case of a weighted bipartite graph.

Our version of the problem cannot be said to either harder or easier than the above problems. One aspect of our problem is harder: we are forced to commit to the set of edges that we query upfront, and are not allowed to adaptively decide which edges to query based on the outcome of the queried edges.[2] The aspect in which it is easier than the other version is that we are allowed to pick a maximum cardinality matching

[2] Note though that the solutions given in previous papers [Chen et al. 2009; Bansal et al. 2012] are non-adaptive.

within the set of edges that exist among the selected edges, whereas the other version forces the algorithm to match edges as they are revealed in case they exist.

Perhaps the more important difference though lies in the type of analysis. While previous work seeks to guarantee a worst-case approximation ratio, or a competitive ratio against an omniscient adversary, we are looking for more practical solutions. Our main result provides an almost optimal solution with high probability for realistic kidney exchange graphs. In this sense our work is closely related to that of Molinaro and Ravi [2011], who study the query-commit problem in kidney exchange graphs, but we believe that the model we use is a better reflection of reality. In addition, we believe that our version of the computational problem more closely mirrors actual kidney exchanges like UNOS.

Several papers study kidney exchanges using realistic random graph models [Ashlagi and Roth 2011; Ashlagi et al. 2012; Toulis and Parkes 2011]. In particular, Ashlagi and Roth [2011] present a compelling model that they use to compare short and long cycles in kidney exchange, and to design matching mechanisms that discourage strategic behavior on the part of hospitals. While the focus of our results is very different, we do use a variant of their model. However, Ashlagi and Roth do not distinguish between the three different compatibility tests, whereas we treat crossmatch tests as fundamentally different.

2. PROBLEM STATEMENT

Given an undirected graph $G = (V, E)$ and a subset of edges $E' \subseteq E$, let $\delta_{E'}(v)$ denote the degree of of $v \in V$ in the subgraph $H = (V, E')$. We consider the following process:

(1) Select a subset of edges $E' \subseteq E$ such that $\delta_{E'}(v) \leq 2$ for all $v \in V$.
(2) Each selected edge $e \in E'$ is revealed to *exist* independently with probability p. Denote the edges in E' that exist by E''.
(3) Compute a maximum cardinality matching $M = M(E'')$ on (V, E'').

In the DOUBLE-CROSSMATCH problem, our goal is to select E' to maximize the expected size of the final matching M. Let us define $\mathit{opt}(G) \triangleq \max_H \mathbb{E}[|M|]$, where $H(V, E')$ is a valid subgraph of G, (i.e., $\forall v \in V$, $\delta_{E'}(v) \leq 2$) and the expectation is taken over the outcomes E'' of E'.

Before moving to describe our main results in detail, we make a couple of easy observations.

OBSERVATION 2.1. *Due to the degree constraints $\delta_{E'}(v) \leq 2$, the subgraph H is a collection of disjoint cycles and paths, and maybe isolated vertices.*

The next observation is that cycles are better than paths.

OBSERVATION 2.2. *A cycle of length $l+1$ has higher expected size of matching than a path of length l (the length of a path or cycle is the number of* edges *in it).*

An easy corollary of the above observation is the following.

COROLLARY 2.3. *If in $H(V, E')$, there exists a path P, whose end points share an edge in $G(V, E)$, then adding that edge to E' does not reduce the size of the expected matching in H.*

3. FOUR-CYCLE COVER AND COMPUTATIONAL COMPLEXITY

The main result of this section is the following theorem, which states that if the graph has a 4-cycle cover[3], then the 4-cycle cover is the *unique* optimal subgraph.

[3] A 4-cycle cover is a collection of cycles each of length 4, such that every vertex lies in exactly one cycle.

THEOREM 3.1. *For any $0 < p < 1$, if the graph G admits a 4-cycle cover, then every optimal H is a 4-cycle cover of G.*

In order to prove the theorem, we rely on the following crucial lemma. We note that Lemma 3.2 holds for *any* non-trivial value of p (i.e., $p \notin \{0, 1\}$).

LEMMA 3.2. *For any $0 < p < 1$, a 4-cycle has strictly higher expected probability of a vertex being matched than a cycle or a path of any other length.*

PROOF. By Observation 2.2, it suffices to show that in a 4-cycle, the average probability of a vertex being matched is strictly higher than that in any other cycle C. Each edge on this cycle exists independently with probability p. Let C_p be the space of outcomes of the edges. Since all edges on a cycle have the same probability of existence p, each vertex in the cycle has the same probability of being matched. We note that — in our analysis — to ensure that each vertex has the same probability of being matched, whenever there is more than one possible maximum matching in an instantiation in C_p, we choose each of the possible maximum matchings with equal probability.

Consider a vertex $v \in C$. Let us calculate the probability that v is matched by breaking up the outcome space into four cases.

(1) Both edges incident to v exist. In this case v is definitely matched if $|C|$ is even (as is the case with a 4-cycle). For odd length cycles, v is matched with probability strictly less than one. This event occurs with probability p^2.
(2) Both edges do not exist. In this case v is definitely not matched, and this occurs with probability $(1-p)^2$.
(3) One of the edges incident to v exists and other does not. Each of these two cases occurs with probability $p(1-p)$.

To calculate the probability that v is matched in the third case, let us look at Figure 1(a) where $v = a_1$ and $n = 6$. The edge (a_1, a_6) is is absent, while the edge (a_1, a_2) is present. Clearly it holds that

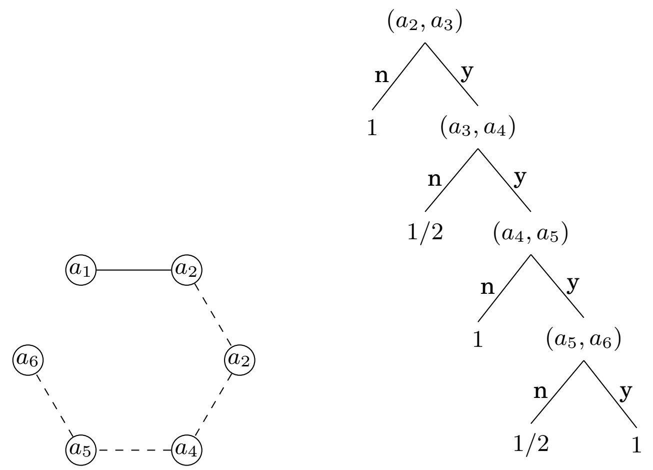
$$\begin{aligned}\Pr[a_1 \text{ matched} | \nexists(a_n, a_1), \exists(a_1 a_2)] =& (1-p)\cdot 1 + p(1-p)\cdot\frac{1}{2} + p^2(1-p)\cdot 1 + p^3(1-p)\cdot\frac{1}{2}\\ &+ \cdots + p^{n-3}(1-p)\cdot f(n) + p^{n-2}\cdot g(n),\end{aligned} \tag{1}$$

where $f(n)$ is 1 if n is odd and $\frac{1}{2}$ if n is even, and $g(n)$ is the opposite. In Equation (1) we have used the observation that if the path starting at a_1 is of even length then a_1 is matched with probability $\frac{1}{2}$, and if it is of odd length then it is matched with probability 1. 1's and $\frac{1}{2}$'s alternate in the above expression; see Figure 1(b) for an illustration. For the case of a 4-cycle, the expression in Equation (1) is equal to $(1-p)\cdot 1 + p(1-p)\cdot\frac{1}{2} + p^2\cdot 1$. For any cycle of length greater than 4, the expression is strictly smaller, because

$$\begin{aligned}&(1-p)\cdot 1 + p(1-p)\frac{1}{2} + p^2(1-p)\cdot 1 + p^3(1-p)\frac{1}{2} + \cdots + p^{n-3}(1-p)\cdot f(n) + p^{n-2}\cdot g(n)\\ &< (1-p)\cdot 1 + p(1-p)\cdot\frac{1}{2} + p^2(1-p)\cdot 1 + p^3(1-p)\cdot 1 + \cdots + p^{n-3}(1-p)\cdot 1 + p^{n-2}\cdot 1\\ &= (1-p)\cdot 1 + p(1-p)\cdot\frac{1}{2} + p^2\cdot 1,\end{aligned}$$

where the inequality is obtained by replacing all the $\frac{1}{2}$'s starting from the fourth term by 1's.

It follows that the expected probability that a vertex is matched is strictly higher in a 4-cycle than in a cycle of length greater than 4. The only other cycle length left to



(a) Dashed edges may not be available.

(b) Probability that vertex v is matched.

Fig. 1. The proof of Lemma 3.2 illustrated for the case of $n = 6$.

consider is 3. A similar analysis shows that the expected probability that a vertex is matched in a cycle of length 3 is

$$(1-p)^2 \cdot 0 + p^2(p \cdot \frac{1}{2} + (1-p) \cdot 1) + 2 \cdot p(1-p) \cdot ((1-p) \cdot 1 + p \cdot \frac{1}{2}) \ ,$$

while for a 4-cycle the expression is

$$(1-p)^2 \cdot 0 + p^2 \cdot 1 + 2 \cdot p(1-p) \cdot ((1-p) \cdot 1 + p(1-p) \cdot \frac{1}{2} + p^2 \cdot 1) \ .$$

It is easy to verify that the 4-cycle expression is strictly greater than the 3-cycle expression for all $0 < p < 1$. □

Lemma 3.2 is one of the main building blocks for our subsequent algorithmic results. We will use it here to establish Theorem 3.1 and that in turn can be applied to establish the computational hardness of our DOUBLE-CROSSMATCH problem.

PROOF OF THEOREM 3.1. Consider a graph G that admits a 4-cycle cover H, and let the subgraph H' be the optimal solution to DOUBLE-CROSSMATCH for G. Assume H' is not a 4-cycle cover of G. In this case, we show that the expected size of matching H is strictly greater than that of H', which contradicts the fact that H' is the optimal solution.

First, it is easy to note that H is a valid solution to the DOUBLE-CROSSMATCH problem since it has at most two edges incident to any node.

The expected size of the matchings of H and H' is the sum over the probability of the vertices being matched in the respective subgraphs. However, Lemma 3.2 states that the average probability of a vertex being matched is highest in a 4-cycle. Furthermore, from Observation 2.1, we know that H' is a collection of cycles and paths. This implies that if H' has any cycle of length other than 4 or a path, then it must have lower expected size of matching than H. This completes the proof. □

3.1. Hardness result

THEOREM 3.3. DOUBLE-CROSSMATCH *is NP-complete.*

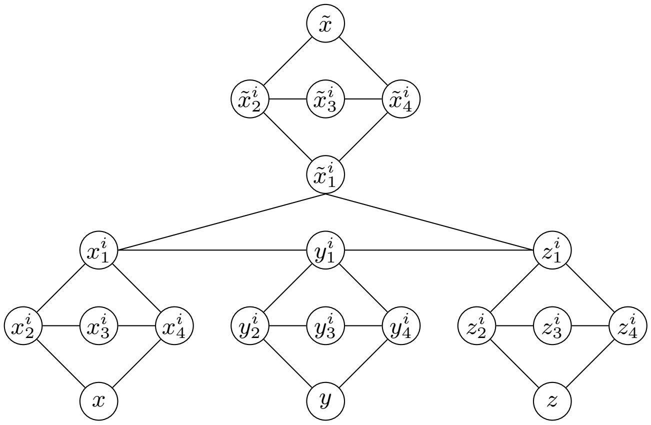


Fig. 2. The gadget used in the proof of Lemma 3.4.

Theorem 3.1 states that if a graph G admits a 4-cycle cover, then the optimal solution to DOUBLE-CROSSMATCH for G is always a 4-cycle cover. Theorem 3.3 therefore follows directly from the following lemma that states that finding whether or not a 4-cycle cover exists is NP hard. The proof of the lemma is similar to the proof that a cover by cycles of length *at most* l for $l \geq 3$ is NP-hard [Abraham et al. 2007, Theorem 1].

LEMMA 3.4. *Deciding whether a graph G admits a cover by 4-cycles is an NP-complete problem.*

PROOF. We reduce the 3D-MATCHING problem to the problem of finding whether a graph admits a 4-cycle cover. In 3D-MATCHING there are three vertex sets X, Y and Z, such that $|X| = |Y| = |Z|$. In addition, we are given a set S of 3-tuples of the form (x, y, z) where $x \in X$, $y \in Y$ and $z \in Z$. The problem is to decide whether there exists a subset $S' \subseteq S$, such that $|S'| = |X| = |Y| = |Z|$ and no two tuples in S' share a vertex in either X or Y or Z. The set S' encodes a perfect matching — every $x \in X$ is matched to a unique $y \in Y$ and $z \in Z$.

For the reduction, we construct a graph G where for every tuple $t_i = (x, y, z)$ in S we introduce the gadget shown in Figure 2. Note that the vertices with superscript i *only* appear in a single gadget – the one corresponding to t_i. The vertices $x, \tilde{x}, y, z$ appear in multiple gadgets, and moreover $\tilde{x}$ appears in *each gadget* that contains x. The intuition is that x is covered if and only if $\tilde{x}$ is covered.

We claim that graph G has a 4-cycle cover if and only if the corresponding 3D-MATCHING problem has a perfect matching. First, if the 3D-MATCHING problem allows a perfect matching, then graph G has a cover through 4-cycles. Indeed, for every tuple $t_i = (x, y, z) \in S'$, we completely cover the corresponding gadget with 4-cycles using only the gadget's vertices (there is only one such cover). For all tuples $t_i = (x, y, z) \in S \setminus S'$, we cover all the vertices except $x, \tilde{x}, y, z$ with 4-cycles using the gadget's vertices. It is easy to verify that this is a complete cover by 4-cycles.

In the other direction, if the graph G has a cover via 4-cycles then the 3D-MATCHING problem admits a perfect matching. The first observation we make is that in a 4-cycle cover for G, for every $x \in X$, the 4-cycle which covers x has to be of the form (x, x_2^i, x_3^i, x_4^i), because the only other possible 4-cycle is (x, x_2^i, x_1^i, x_4^i) but now x_3^i cannot be covered. In addition, once for a particular i the 4-cycle (x, x_2^i, x_3^i, x_4^i) is included the corresponding x_1^i can only be covered through the 4-cycle $(x_1^i, y_1^i, z_1^i, \tilde{x}_1^i)$. This in turn implies that we must completely cover the gadget using only the gadget's ver-

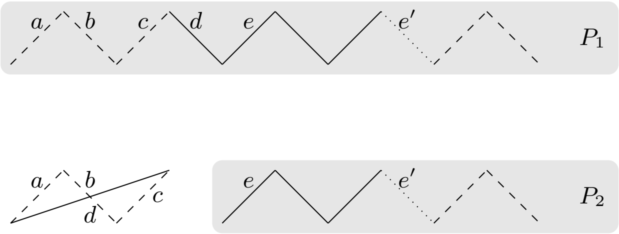


Fig. 3. The proof of Claim 4.2 illustrated for the case of $l = 10$. Solid edges exist, dotted edges do not exist, and dashed edges may or may not exist.

tices. For every i such that $(x_1^i, y_1^i, z_1^i, \tilde{x}_1^i)$ is included in the cover for graph G, the tuple (x, y, z) is included in the set S'. It is easy to verify that S' encodes a solution to the 3D-MATCHING problem. □

3.2. Complete Graphs

Another corollary which follows from Theorem 3.1 is the following.

COROLLARY 3.5. *Consider a complete graph $G(V, E)$, i.e., $E = \{(u, v) : u \neq v, u, v \in V\}$, such that $|V|$ is divisible by 4. Then the optimal subgraph H of G is composed of $|V|/4$ vertex disjoint 4-cycles.*

PROOF. From Theorem 3.1, we know that if graph G admits a 4-cycle cover then every optimal subgraph of H is a 4-cycle cover. The complete graph $G(V, E)$ with $|V|$ divisible by 4 does admit a cover through 4-cycles. □

4. BIPARTITE GRAPHS

Our next goal is to characterize optimal solutions for complete bipartite graphs. Note that since a bipartite graph may not admit a 4-cycle cover, from the results so far we do not know what an optimal collection of edges for a bipartite graph looks like. The main result for this section is the following.

LEMMA 4.1. *Consider a complete bipartite graph $G(L \cup R, L \times R)$, with $|L| \leq |R| \leq 2|L|$. Then there exists an optimal solution H for graph G that consists only of 4-cycles, paths of length 2, and at most one path of length 4 or a single edge.*

As opposed to complete graphs, where a cover by 4-cycles is uniquely optimal if one exists, here we are not claiming that this is the unique optimal subgraph. For our purposes, the aspect of the lemma which will prove crucial later is that only "small" structures are required. To prove this lemma, we first show that we do not lose anything by restricting our attention to "short" paths.

CLAIM 4.2. *For any $l \geq 6$, the expected size of the matching under a 4-cycle plus a path of length $l - 4$ is at least the expected size of the matching under a path of length l.*

PROOF. Let P_1 be a path of length $l \geq 6$, and call its first four edges from the left a, b, c, d, e. Now we use the first four vertices to close a cycle, and we also call its edges a, b, c, d; the remaining path of length $l - 4 \geq 2$, which starts with e, is denoted by P_2. We first make the observation that in any instantiation of the edges, if the edge d is absent then the two structures have an equal number of matched edges. Hence we only consider outcomes where edge d is present.

Consider an instantiation of the edges (such that edge d is present) and let edge e' be the first edge in P_2, starting from edge e and going to the right, that fails (if no such edge e' exists, let $e' = \phi$, i.e., null). See Figure 3 for an illustration. To the right of e',

Table I. The table shows the difference in the size of matching between 4-cycle plus path P_2', and path P_1' for various possibilities of edge outcomes of a, b, c and whether $|P_2'|$ is even or odd. Edge d exists in all cases. An edge exists (resp., does not exist) if its column shows 1 (resp., 0).

a	b	c	$\lvert M(4C+P_2')\rvert - \lvert M(P_1')\rvert$		a	b	c	$\lvert M(4C+P_2')\rvert - \lvert M(P_1')\rvert$	
			Even	Odd				Even	Odd
0	0	0	0	+1	1	0	0	-1	0
0	0	1	0	0	1	0	1	0	0
0	1	0	0	+1	1	1	0	0	+1
0	1	1	0	+1	1	1	1	0	0

both paths P_1 and P_2 have the same number of matched edges. Hence, we only need to look to the left of e'; denote the path segments of P_1 and P_2 to the left of e' as P_1' and P_2' respectively.

We now look at all possible outcomes of edges a, b and c, and the length of the path P_2'. We tabulate our observations in Table 4; they are easy to verify. Using the table, if $\mathbb{E}[|M(4C + P_2')|]$ and $\mathbb{E}[|M(P_1')|]$ denote the expected number of matched edges for the cases of 4-cycle plus path P_2', and path P_1' respectively, then

$$\begin{aligned}&\mathbb{E}[|M(4C + P_2')|] - \mathbb{E}[|M(P_1')|]\\ &= p\big[((1-p)^3 + (1-p)^2p + (1-p)p^2 + p^2(1-p))\mathrm{Pr}(|P_2'| \text{ odd}) - p(1-p)^2\mathrm{Pr}(|P_2'| \text{ even})\big]\\ &= p(1-p)\cdot\big[(1-p+2p^2)\cdot \mathrm{Pr}(|P_2'| \text{ odd}) - p(1-p)\cdot \mathrm{Pr}(|P_2'| \text{ even})\big],\end{aligned}$$

where in the first equality the leftmost factor of p stands for the probability of edge d being present, the term $((1-p)^3 + (1-p)^2p + (1-p)p^2 + p^2(1-p))$ sums up the probabilities of the outcomes of edges a, b, c where $4C + P_2'$ has one more matched edge than P_1', and the term $p(1-p)^2$ is for the single case where P_1' has one more matched edge than $4C + P_2'$. Now, with $l' = l - 4$,

(1) l is odd: $\mathrm{Pr}(|P_2'| \text{ odd}) = (1-p)\cdot\big(p + p^3 + p^5 + \cdots + p^{l'-2}\big) + p^{l'}$ and $\mathrm{Pr}(|P_2'| \text{ even}) = (1-p)\cdot(1 + p^2 + p^4 + \cdots + p^{l'-1})$. So, we have $\mathrm{Pr}(|P_2'| \text{ odd}) \geq p \cdot \mathrm{Pr}(|P_2'| \text{ even})$. And hence,
$$(1-p+2p^2)\cdot \mathrm{Pr}(|P_2'| \text{ odd}) \geq p(1-p)\cdot \mathrm{Pr}(|P_2'| \text{ even})\ .$$

(2) l is even: $\mathrm{Pr}(|P_2'| \text{ odd}) = (1-p)\cdot\big(p + p^3 + p^5 + \cdots + p^{l'-3} + p^{l'-1}\big)$ and $\mathrm{Pr}(|P_2'| \text{ even}) = (1-p)\cdot(1 + p^2 + p^4 + \cdots + p^{l'-2}) + p^{l'}$. So, we have $\mathrm{Pr}(|P_2'| \text{ odd}) = p\cdot\mathrm{Pr}(|P_2'| \text{ even}) - p^{l'+1}$. And hence,
$$(1-p+2p^2)\cdot\mathrm{Pr}(|P_2'| \text{ odd}) - p(1-p)\cdot\mathrm{Pr}(|P_2'| \text{ even}) = 2p^3\cdot\mathrm{Pr}(|P_2'| \text{ even}) - (1-p+2p^2)\cdot p^{l'+1}\ .$$
But, $\mathrm{Pr}(|P_2'| \text{ even})$ is at least $(1-p) + p^{l'}$ from the first expression that we wrote for that quantity. It follows that $2p^3 \cdot \mathrm{Pr}(|P_2'| \text{ even}) \geq (1-p+2p^2)\cdot p^{l'+1}$ since $l' \geq 2$ as $l \geq 6$.

Therefore, $\mathbb{E}[|M(4C + P_2')|] \geq \mathbb{E}[|M(P_1')|]$ and since $\mathbb{E}[|M(4C + P_2)|] - \mathbb{E}[|M(P_1)|] = \mathbb{E}[|M(4C + P_2')|] - \mathbb{E}[|M(P_1')|]$, the lemma follows. □

Claim 4.2 directly implies that in a complete bipartite graph, paths of length at least seven are useless. Next we compare 4-cycles and short paths; we defer the proof of the next claim to Appendix D.

CLAIM 4.3.

(*1*) *For any even $l \geq 4$ and any $p \in (0, 1)$, the probability of a vertex being matched in a cycle of length l is strictly more than that in a cycle of length $l + 2$.*

(*2*) *For any $p \in (0,1)$, the expected number of matched edges in a 4-cycle plus an edge is strictly more than the expected number of matched edges in a cycle of length 6.*
(*3*) *The expected number of matched edges in a 4-cycle plus two paths of length 2 is equal to the expected number of matched edges in two paths of length 4.*

We now present the proof of Lemma 4.1.

PROOF OF LEMMA 4.1. Consider an optimal choice of edges O for the complete bipartite graph G. If O contains any paths of odd length, we can increase the quality of the solution by adding an edge between the end points to get cycles of even length. In addition, cycles of odd length are impossible. Hence an optimal solution can contain only cycles of even length and paths of even length. Using the first two parts of Claim 4.3 we can assume that all cycles are of length 4.

Next, by repeated application of Claim 4.2, we can convert O to a solution O', where we are left with only 4-cycles and paths of length 1 and 4, and the expected number of matched edges in O' is at least as large as O. Multiple paths of length 1 can be combined into one longer even-length path (plus maybe a path of length 1), and then further converted to 4-cycles plus maybe a path of length 4 (and plus maybe a path of length 1).

If at this stage there is more than one path of length 4 in O', we can use part 3 of Claim 4.3 to further prune these paths and replace them with 4-cycles and paths of length 2. At this point, we can have at most one path of length 4 and at most one path of length 1. But this pair of structures is clearly worse than a path of length 6, which by Claim 4.2 is worse than a 4-cycle plus a path of length 2. □

5. GENERAL GRAPHS

Having discussed the case of complete graphs (Section 3.2) and bipartite graphs (Section 4), we now move our attention to general graphs. The following lemma states that if there exists a vertex u which does not have any edge incident to it in the subgraph H, but which has an edge incident to it in the original graph G, then that edge can be included in the subgraph H (perhaps requiring some other edge in H to be deleted), without decreasing the expected size of matching of H. Its proof is relegated to Appendix B.

LEMMA 5.1. *(No vertex left behind.) Consider an undirected graph $G(V, E)$, and a subgraph $H(V, E')$ ($E' \subseteq E$) with $\delta_{E'}(v) \leq 2$. Suppose there exists a vertex $u \in V$ with $\delta_{E'}(u) = 0$ but $\delta_E(u) > 0$. Let v be a vertex which has an edge with u under E. Then we can add the edge (u, v) to E', and if needed, remove some other edge incident to v under E' in order to ensure $\delta_{E'}(v) \leq 2$, without reducing the expected size of matching of E'.*

From Lemma 5.1, we can infer the following result.

COROLLARY 5.2. *There exists an optimal solution $H(V, E')$ for the subgraph of $G(V, E)$ with the following property. For every vertex u that has $\delta_{E'}(u) = 0$,*

(*1*) *either $\delta_E(u) = 0$,*
(*2*) *or for every edge (v, u) present in E, $\delta_{E'}(v) = 2$, and if b and d are the two vertices adjacent to v under E', then $\delta_{E'}(b) = 1 = \delta_{E'}(d)$.*

The proof of the corollary is straightforward: If there exists a vertex u for which the stated property is violated then we can apply Lemma 5.1 and convert the solution to one where it is not, without decreasing the expected maximum matching size.

6. COMPLETE KIDNEY EXCHANGE GRAPHS

In this section, we will deal with a kidney exchange graph where every pair of vertices that are blood-type compatible share an edge. In our results in this section we implicitly assume that tissue typing tests are always successful; this assumption is relaxed in Section 7.

There are four blood types A, B, AB, and O. For blood-type compatibility the patient should have as many types of antigens as the donor. Blood type O indicates absence of antigens and hence a donor of blood type O is blood-type compatible with all other blood groups. Blood groups A, B, and AB indicate presence of antigens A, B, and both A and B, respectively. Hence, a donor with blood type A is blood type compatible with a patient of either blood type A or AB. A patient with blood type AB is blood type compatible with a donor of any blood group.

Since every node in the graph represents a (patient, donor) pair, we can label each node by the blood-types of the patient and the donor. For instance, if the patient has blood type A and the donor blood type AB, then the label is $A - AB$.

We now borrow some definitions from Ashlagi and Roth [2011] that will help our presentation. In each definition $X, Y \in \{A, B, AB, O\}$.

Definition 6.1.

(1) A label $X - Y$ is *over-demanded* if $X \neq Y$ and Y is blood-compatible to donate to X.
(2) A label $X - Y$ is *under-demanded* if $X \neq Y$ and X is blood-compatible to donate to Y.
(3) All labels of the form $X - X$ are known as *self-demanded*.
(4) The pair of labels $A - B$ and $B - A$ constitute *reciprocally-demanded* types.

Note that if $X - Y$ is over-demanded, then $Y - X$ *must be* under-demanded. We will make the following assumption: For every $X - Y$ such that $X - Y$ is over-demanded and $Y - X$ is under-demanded, the number of nodes in the graph with label $X - Y$ is less than half the number of nodes with label $Y - X$. For instance, an implication of this assumption is that the number of nodes with blood type $AB - A$ is *less than half* of the number of nodes with blood-type $A - AB$.

Why might such an assumption be realistic? The justification stems from the way patient-donor pairs are formed in practice. Observe that every patient-donor pair that is not blood-type compatible has to enter the kidney exchange pool. On the other hand, if the donor is blood-type compatible to donate to the patient, then only pairs who fail a tissue typing or crossmatch test join the pool. Hence, a priori one has reason to believe that the number of pairs in the kidney exchange pool that have label $X - Y$ is significantly smaller than the number of pairs with label $Y - X$, so for example Roth et al. [2007] assume that there is an endless pool of underdemanded pairs. Moreover, often the willing donor is a family member of the patient, and among family members there is a higher chance of the tissue typing and crossmatch tests being successful. In fact, the factor $1/2$ has been used by Ashlagi and Roth [2011], who based this assumption on real data [Zenios et al. 2001].

Now, let us the consider the reciprocally demanded labels $A - B$ and $B - A$. Note that a donor with blood-type A cannot donate to a patient with blood-type B, and vice versa. Hence, every (patient, donor) pair of either of these types is forced to enter the kidney exchange market. Moreover, the chances of (patient, donor) pair having blood type $A - B$ is the same as them having $B - A$, since there is no reason to believe that a person with blood type A has a higher or lower chance of kidney failure than a person of type B. Hence, in our complete kidney exchange graph, we assume that the number of nodes with label $A - B$ is *approximately* the same as those with label $B - A$.

With this we are ready to define our model of the complete kidney exchange graph, where for now (until Section 7) we only consider blood-type compatibility and ignore tissue-type compatibility.

Definition 6.2. A *complete kidney graph* is a graph $G(V, E)$ with the following properties. The vertex set V can be partitioned into the sets V_{X-Y} where X and Y are the blood types of the patient and the donor respectively ($X, Y \in \{A, B, AB, O\}$). Furthermore,

(1) Every pair of vertices in G that are *blood-type compatible* share an edge.
(2) For each over-demanded label $X - Y$, $|V_{X-Y}| < \frac{1}{2}|V_{Y-X}|$.
(3) The reciprocally demanded labels obey $\frac{1}{2}|V_{B-A}| \leq |V_{A-B}| \leq 2 \cdot |V_{B-A}|$.

We define the term *an almost optimal subgraph* to denote a subgraph whose expected matching size is off from the optimal solution only by constant additive factors.

Definition 6.3. An *almost optimal subgraph* H for a graph G is a solution to the DOUBLE-CROSSMATCH problem for G, which has expected size of matching at least $opt(G) - O(1)$.

We now present the structure of an almost optimal solution for the complete kidney exchange graph (see Figure 4 for an illustration).

THEOREM 6.4. *The subgraph $H(V, E')$ with the following description is an almost optimal subgraph for the complete kidney exchange graph $G(V, E)$.*

(*1*) (Self-demanded form 4-cycles among themselves) *For every self-demanded label $X - X$, the edges of H constitute a 4-cycle cover of all (but for maybe $O(1)$) vertices of that label.*
(*2*) (Each over-demanded pairs with two under-demanded) *For every pair of over-demanded $(X - Y)$ and under-demanded (Y − X) labels, every node with label $X - Y$ has two edges incident to a unique pair of vertices with label $Y - X$.*
(*3*) (Reciprocally demanded pair) *Every node in $A - B$ is involved in either a 4-cycle with one vertex of its own label and two nodes of the opposite label (i.e., $B - A$), or a path of length two using vertices of the opposite label and maybe of its own label. A similar statement holds for each node in $B - A$.*

The crucial result that helps us to prove the the optimality of the above solution is the following lemma, whose proof we defer to Appendix A. In a sense, it distills the core properties of kidney exchange graphs, and presents the structure of an optimal solution for all graphs that have these properties.

Definition 6.5. An undirected graph $G(V, E)$ is said to be *lopsided-bipartite partitionable* if it has the following structure. The vertex set V can be partitioned into k pairs of sets (P_i, Q_i) ($1 \leq i \leq k$) and R for some k, such that $V = \bigcup_{i=1}^{k}(P_i \cup Q_i) \bigcup R$. Furthermore, for each $1 \leq i \leq k$,

(1) $|Q_i| > 2 \cdot |P_i|$
(2) P_i and Q_i form a complete bipartite graph.
(3) No vertex $v \in Q_i$ has an edge incident to it from any vertex in $R \cup \bigcup_{j=1}^{k} Q_j$.

All other possible edges may or may not be present in G.

LEMMA 6.6. *For a lopsided-bipartite partitionable graph $G(V, E)$ with $V = \bigcup_{i=1}^{k}(P_i \cup Q_i) \bigcup R$, as in Definition 6.5, there exists an optimal subgraph $H(V, E')$ with the property that for every $1 \leq i \leq k$, all vertices $v \in P_i$ have two edges incident to a*

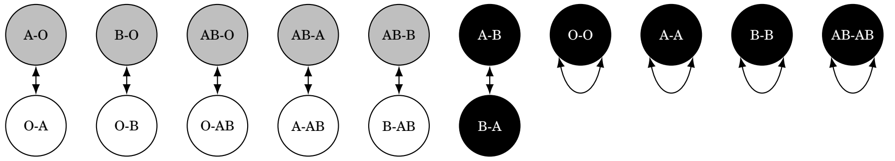


Fig. 4. Edges chosen by Algorithm 6.4 in the kidney exchange graph. The grey circles are over-demanded labels, the white circles are under-demanded labels, and the black circles are reciprocally demanded and self-demanded labels.

unique pair of vertices in $Q_i \times Q_i$. In particular, H does not have any edge between a vertex in P_i (for any i) and a vertex in R.

We now complete the proof of the main result.

PROOF OF THEOREM 6.4. We first set the stage for the application of Lemma 6.6. Consider the following settings of P_i's, Q_i's and R.

$(P_1, Q_1) \triangleq (V_{AB-A}, V_{A-AB}), \quad (P_3, Q_3) \triangleq (V_{AB-O}, V_{O-AB}) \quad (P_5, Q_5) \triangleq (V_{B-O}, V_{O-B})$
$(P_2, Q_2) \triangleq (V_{AB-B}, V_{B-AB}), \quad (P_4, Q_4) \triangleq (V_{A-O}, V_{O-A}),$
$R \triangleq (V_{A-A}) \cup V_{B-B} \cup V_{O-O} \cup V_{AB-AB} \cup V_{A-B} \cup V_{B-A})$

Every over-demanded label with the corresponding under-demanded label has been put in one of the (P_i, Q_i)'s with the over-demanded label taking the place of P_i. The set of self-demanded and reciprocally demanded labels have been put in R. Looking at Definition 6.2 and Table II to infer the edges present in G, we can see the graph G satisfies the condition to apply Lemma 6.6.

Hence, using Lemma 6.6, we know that there exists an optimal solution $K(V, E')$ for the complete kidney exchange graph G, that for every over-demanded/under-demanded pair of labels, satisfies the property that every vertex of the over-demanded label $(X - Y)$ has two edges incident to a unique pair of vertices of the under-demanded label $(Y - X)$.

Furthermore, in graph G (and hence in graph K), the vertices of the under-demanded types do not have an edge to any vertex in the set R as defined above.

From Table II, it is easy to see that

(1) For every self-demanded label $X - X$, a vertex of that label has edges in graph G to either vertices of of its own label or to an over-demanded label.
(2) Vertices labeled $A - B$ (resp., $B - A$) share edges in graph G with vertices with either an overdemanded label or label $B - A$ (resp., $A - B$).

By Lemma 6.6, the optimal subgraph K does not have any edges between a vertex with an over-demanded label and a vertex with either a self-demanded or reciprocally demanded label. Hence,

(1) For every self-demanded labeled $X - X$ vertex, graph K can only include edges that are incident to the vertex from other vertices of the same label.
(2) For a vertex labeled $A - B$ (resp., $B - A$), graph K can only include edges that are incident to it from vertices with label $B - A$ (resp., $A - B$).

In other words, for each self-demanded label $X - X$, graph K might as well treat the complete graph formed by the vertices of that label in graph G as a separate entity and optimize on it. Similarly, graph K can optimize over the bipartite graph formed by the vertices of the reciprocally-demanded labels $A - B$ and $B - A$ separately.

For the complete graph formed by the vertices of a self-demanded labeled $X - X$, we know that if $|V_{X-X}|$ is divisible by 4, Lemma 3.1 states that the optimal solution is a 4-cycle cover of V_{X-X}. Otherwise, a set of vertex disjoint 4-cycles that cover all but $O(1)$ of the vertices is an almost optimal solution for the complete graph V_{X-X} (for sake of analysis, we can throw out $O(1)$ vertices to get a complete graph whose number of vertices is divisible by 4, and we know that for this remaining graph, the 4-cycle cover is optimal).

Similarly, applying Lemma 4.1 to the bipartite graph formed by vertices with the reciprocally demanded labels $A - B$ and $B - A$, we know that an optimal solution consists of a cover of the vertices by 4-cycles, paths of length two and at most one path of length 4 or an edge. If we throw out this one of path of length 4 or the edge, we get an almost optimal solution consisting purely of 4-cycles and paths of length two. If an $A - B$ vertex is in a 4-cycle, then it shares this 4-cycle with one $A - B$ vertex and two $B - A$ vertices. Moreover, depending on whether $|A - B| \geq |B - A|$ or the other way, any path of length two will contain two vertices of label $A - B$ and one of $B - A$ or vice-versa respectively.

Hence, the graph H as described in the statement of the theorem will have an expected size of matching at least that of K minus $O(1)$. We lose $O(1)$ terms if a 4-cycle cover for any of the complete graphs formed by vertices of a self-demanded label is not possible or if the bipartite graph formed by $A - B$ and $B - A$ cannot be covered using 4-cycles and paths of length 2. This completes the proof. □

Table II. The set of compatible blood-types for all under-demanded, self-demanded and reciprocally-demanded type vertices.

Patient-Donor	Com. Patient	Com. Donor
A-A	A/AB	O/A
B-B	B/AB	O/B
O-O	O/A/B/AB	O
AB-AB	AB	O/A/B/AB
A-B	B/AB	O/A
B-A	A/AB	O/A
A-AB	AB	O/A
B-AB	AB	B/AB
O-A	A/AB	O
O-B	B/AB	O
O-AB	AB	O

An easy corollary of Theorem 6.4 is the following result.

COROLLARY 6.7. *There exists an almost optimal solution $H(V, E')$ for the complete kidney exchange graph G with the following properties:*

(*1*) *For each self-demanded label, there are $\lfloor |V_{X-X}| \rfloor / 4$ vertex-disjoint cycles of length 4 in the subgraph H.*

(*2*) *For each over-demanded label $X - Y$, there are $|V_{X-Y}|$ vertex-disjoint paths of length 2, each path involving a vertex of label $X - Y$ with an edge incident to two unique vertices of label $Y - X$.*

(*3*) *There are $\lfloor y \rfloor$ vertex disjoint paths of length 2 and $\lfloor z \rfloor$ vertex disjoint cycles of length 4 where x and y are given by the equations*

$$y + 2z = \min(|V_{A-B}|, |V_{B-A}|) \tag{2}$$

$$2y + 2z = \max(|V_{A-B}|, |V_{B-A}|) \tag{3}$$

In each such path of length 2, a vertex of label $\arg\min(|V_{A-B}|, |V_{B-A}|)$ has an edge each to two vertices of the other label. Every cycle of length 4 has two vertices of label $A - B$ that share an edge each with two vertices of label $B - A$.

7. REALISTIC KIDNEY EXCHANGE GRAPHS

We now remove the assumption of successful tissue typing tests that we made in Section 6. In practice, if two pairs of donor-patient are blood-type compatible then the tissue-type test succeeds with some constant probability [Ashlagi and Roth 2011]. This probability depends on biological parameters of the patients and donors. Hence, a realistic kidney exchange graph can be seen as drawn from a distribution over graphs, where the distribution is defined as follows: The vertex set of each graph in the distribution is the same as the complete kidney exchange graph (Definition 6.2), and obeys the constraints imposed on its various vertex sets. Each edge of the complete kidney exchange graph exists independently in the randomly drawn graph with a constant probability c (which for our purposes can be thought of as a lower bound). The approach of drawing a realistic kidney exchange graph from a similar distribution has been taken before by Ashlagi and Roth [2011] and Toulis and Parkes [2011].

ALGORITHM 1: Polynomial time algorithm for the DOUBLE-CROSSMATCH problem for realistic kidney exchange graphs.

```
Input: A realistic kidney exchange graph $G_r$ drawn from a distribution.
Output: A subgraph $H_r$ of $G_r$ that is a solution to the DOUBLE-CROSSMATCH problem for $G_r$.
(1) For each of the complete graphs $V_{A-A}$, $V_{B-B}$, $V_{AB-AB}$, $V_{O-O}$, we run Algorithm 2 and add
    to $H_r$ the edges it returns.
(2) For each of the bipartite graphs $(V_{AB-A}, V_{A-AB})$, $(V_{AB-B}, V_{B-AB})$, $(V_{AB-O}, V_{O-AB})$,
    $(V_{B-O}, V_{O-B})$, $(V_{AB-O}, V_{O-AB})$, we run Algorithm 3 and add to $H_r$ the edges it returns.
(3) For the bipartite graph $(V_{A-B}, V_{B-A})$, we run Algorithm 4 and add to $H_r$ the edges it
    returns.
```

ALGORITHM 2:

```
Input: A random graph $G_r$ drawn from a complete graph $G$.
Output: A subgraph $H_r$ of $G_r$ with every node having at most incident edges.
(1) Throw out $O(1)$ vertices from $G_r$ to make the cardinality of the vertex-set divisible by 4.
(2) Uniformly randomly partition the vertices of $G_r$ into two sets $A$ and $B$ with $|A| = |B|$.
(3) In $A$, pair up the vertices uniformly randomly to get a set $A'$ which treats each pair as a
    vertex and hence $|A'| = |A|/2$. The vertices of $A'$ can denoted as $v_{xy}$ where $x$ and $y$ are the
    two vertices in $A$ that were paired up. Do a similar operation with $B$ to get $B'$.
(4) Introduce an edge between a vertex $v_{xy}$ in $A'$ and a vertex $v_{x',y'}$ in $B'$ if $G_r$ contains all the
    edges $(x, x'), (x, y'), (y, x'), (y, y')$. Note that if $G_r$ contains all these edges, then $x, x', y, y'$
    form a 4-cycle in $G_r$.
(5) Compute the maximum matching $M$ in the bipartite graph formed between $A'$ and $B'$.
(6) For each edge $(v_{xy}, v_{st})$ included in $M$, include the corresponding 4-cycle $(x, s, y, t)$ in $H_r$.
```

ALGORITHM 3:

```
Input: A random graph $G_r$ drawn from a lopsided complete bipartite graph $G(A \cup B, E)$, with
    $|A| < \frac{1}{2}|B|$.
Output: A subgraph $H_r$ of $G_r$ where each vertex is incident to at most two edges.
(1) Randomly pair up the vertices in $B$ (if $|B|$ is not divisible by 2, throw out a vertex from $B$
    and then pair up the remaining vertices). Construct a new set $B'$ by introducing a vertex
    $v_{xy}$ in $B$ for each pair $(x, y)$ of vertices created from $B$.
(2) Construct a bipartite graph $G'$ between $A$ and $B'$. Introduce an edge between a vertex
    $u \in A$ and a vertex $v_{xy} \in B'$, if the pair of edges $(u, x)$ and $(u, y)$ exist in $G$.
(3) Find a maximum matching $M$ in the bipartite graph $G'$.
(4) For every matched edge $(u, v_{xy})$ in $M$, add the edges $(u, x)$ and $(u, y)$ to $H_r$.
```

```
ALGORITHM 4:
Input: A random graph $G_r$ drawn from an almost balanced complete bipartite graph
    $G(L \cup R, E)$, with $|L| \leq |R| \leq 2|L|$.
Output: A subgraph $H_r$ of $G_r$ where each vertex is incident to at most two edges.
(1) With the given values of $|L|$ and $|R|$, solve for $x$ and $y$ in the equations $2 \cdot x + y = |L|$ and
    $2 \cdot x + 2 \cdot y = |R|$. Consider disjoint subsets $L_1$ and $L_2$ of $L$ of sizes $2 \cdot \lfloor x \rfloor$ and $y$ respectively.
    Similarly, consider disjoint subsets $R_1$ and $R_2$ of $R$ of sizes $2 \cdot \lfloor x \rfloor$ and $2 \cdot y$ respectively.
(2) Pair up the vertices in $L_1$ and for every such pair $(s, t)$, introduce a vertex $v_{st}$ in a new set
    $L'_1$. Similarly, pair up vertices in $R_1$ and $R_2$ to construct sets $R'_1$ and $R'_2$ respectively.
(3) Construct bipartite graphs $G_1$ over $L'_1 \cup R'_1$, and introduce an edge between vertices
    $v_{st} \in L'_1$ and $v_{pq} \in R'_1$ in $G_1$, if each of the edges $(s,p)$, $(p,t)$, $(t,q)$ and $(q,s)$ are present in
    $G_r$ (i.e., the vertices $(s,p,t,q)$ form a 4-cycle in $G_r$).
(4) Construct bipartite graph $G_2$ over $L_2 \cup R'_2$, and introduce an edge between vertices $u \in L_2$
    and $v_{st} \in R'_2$ if the edges $(u,s)$ and $(u,t)$ exist (i.e., $(s,u,t)$ form a path of length 2) in $G_r$.
(5) Find a maximum-cardinality matching $M_1$ in $G_1$, and $M_2$ in $G_2$.
(6) For every edge $(v_{st}, v_{pq}) \in M_1$, include the edges of the 4-cycle $(s,p,t,q)$ in $H_r$. For every
    edge $(u, v_{st}) \in M_2$, include the edges of the path of length 2 formed by $(s,u,t)$ in $H_r$.
```

We now present our main result, building on most of the results presented above.

THEOREM 7.1. *For a randomly drawn graph G_r from the kidney exchange graph G, we can algorithmically find in polynomial time a subgraph H_r that with probability at least $1-o(\frac{1}{opt(G)})$ has expected matching size at least $(1-o(1))opt(G) \geq (1-o(1))opt(G_r)$.*

PROOF. From the characterization of an almost optimal subgraph H for the kidney graph G as mentioned in Lemma 6.7, we know that H will have the following:

(1) $\alpha \triangleq |V_{AB-A}| + |V_{AB-B}| + |V_{AB-O}| + |V_{B-O}| + |V_{AB-O}| + \lfloor y \rfloor$ many paths of length 2
(2) $\beta \triangleq \lfloor (|V_{A-A}| + |V_{B-B}| + |V_{AB-AB}| + |V_{O-O}|)/4 \rfloor + \lfloor z \rfloor$ many cycles of length 4

where y and z are given by the set of equations

$$y + 2z = \min(|V_{A-B}|, |V_{B-A}|) \tag{4}$$

$$2y + 2z = \max(|V_{A-B}|, |V_{B-A}|) \tag{5}$$

Hence the expected size of matching of H is given by $\alpha \cdot M_{2P} + \beta \cdot M_{4C}$ where M_{2P} and M_{4C} denote the expected size of matching in a path of length 2 and a cycle of length 4 respectively.

We will show that for a random graph G_r, with high probability, we can algorithmically find a subgraph H_r of G, that is composed of $\alpha - o(n)$ many paths of length 2 and $\beta - o(n)$ many cycles of length 4. Hence, with high probability, the expected matching size of H_r would be $(\alpha - o(n)) \cdot M_{2P} + (\beta - o(n)) \cdot M_{4C} = opt(G) - o(n)$. Here $n = |V|$. It is easy to see that $opt(G) \geq opt(G_r$ for all graphs G_r since the edge set of G_r is a subset of that of G, and hence the optimal subgraph solution of G_r is also a subgraph of G. Hence, it also follows that the expected matching size of H_r is $opt(G_r) - o(n)$.

All that is left to prove is that for a random graph G_r, with high probability, we can algorithmically find a subgraph H_r of G, that is composed of $\alpha - o(n)$ many paths of length 2 and $\beta - o(n)$ many cycles of length 4. We claim Algorithm 1 has the desired properties.

The algorithm can be easily seen to run in polynomial since each of the sub-algorithms clearly runs in polynomial time. We now complete the analysis. Some of the claims and proofs are deferred to the appendix.

(1) For each of the bipartite graphs $(V_{X-Y}, V_{Y-X}) \in \{(V_{AB-A}, V_{A-AB}), (V_{AB-B}, V_{B-AB}), (V_{AB-O}, V_{O-AB}), (V_{B-O}, V_{O-B}), (V_{AB-O}, V_{O-AB})\}$, using Claim C.7, we add to H_r, with probability at least $1 - o(\frac{1}{|V_{X-Y}|})$, $|V_{X-Y}|$ many paths of length 2.

(2) For each of the complete graphs $V_{X-X} \in \{V_{A-A}, V_{B-B}, V_{AB-AB}, V_{O-O}\}$, applying Claim C.6, we add, with probability at least $1 - o(\frac{1}{V_{X-X}})$, $\lfloor(|V_{X-X}|/4\rfloor - O(1)$ many 4-cycles to H_r.

(3) For the bipartite graph (V_{A-B}, V_{B-A}), we can infer from Claim C.8, that we add to H_r, with probability at least $1 - o(\frac{1}{T})$, $\lfloor y \rfloor - o(T)$ many paths of length 2 and $\lfloor z \rfloor - o(T)$ many cycles of length 4, where $T = |V_{A-B} \cup V_{B-A}|$.

We now need to sum up over the probability of failure in each of the high probability statements given above. For each high probability statement given above, either the probability of failure is $o(\frac{1}{opt(G)})$ or the contribution of that term to the size of optimal matching $opt(G)$ is $o(opt(G))$.

We only have a small number of sub-algorithms, hence using the union bound we can say that with probability at least $1 - o(\frac{1}{opt(G)})$, the size of expected matching of the graph H_r returned by the algorithm is $(1 - o(1))opt(G)$. □

8. DISCUSSION

Taking an algorithmic point of view, our paper focuses on a special rather than a general problem. In particular, we only consider the case where the collection of selected edges E' includes at most two edges per vertex. There are two reasons for this restriction. The first is that even the extension from two to three is extremely difficult, because the relevant structures in the latter setting are no longer just cycles and paths. The second reason is practical: current kidney exchanges use only one crossmatch per matched vertex; tweaking the existing policy to allow two crossmatches seems realistic in terms of additional costs. It is unclear whether performing more than two crossmatches is feasible in practice.

Although we aim for a realistic kidney exchange model, it does differ from reality in several important ways:

— Some kidney exchanges include altruistic donors that initiate a *chain* of donations; very few existing theoretical models deal with chains [Dickerson et al. 2012b; Ashlagi et al. 2012].

— While our model of kidney exchanges is static, in reality a matching is computed on a weekly or monthly basis, and over time patients and donors arrive and depart. Several recent papers consider the *dynamics* of kidney exchange [Ünver 2010; Dickerson et al. 2012a].

— In addition to pairwise exchanges, real kidney exchanges match along 3-cycles. Although this extension does not require any additional modeling, technically it is very challenging because the optimal structures are much harder to characterize. Note that papers on the query-commit problem also focus on pairwise exchanges [Chen et al. 2009; Bansal et al. 2012; Costello et al. 2012].

— In practice kidney exchanges weight edges according to the quality of the fit (for example an edge between an old donor and young patient would have low weight). While weights are not taken into account in most existing kidney exchange papers (see, e.g., [Ashlagi et al. 2010; Toulis and Parkes 2011; Ashlagi and Roth 2011; Caragiannis et al. 2011; Ashlagi et al. 2012]), they do play a role in the recent work of Bansal et al. [2012]. A related extension has to with assigning a different probability of crossmatch failure p_e to each edge $e \in E$ [Chen et al. 2009; Bansal et al. 2012; Costello et al. 2012].

Nevertheless, as simplified as our theoretical model is, we believe that our results indicate that the concept of performing multiple crossmatch tests could be practical. Indeed, we expect a shift to two crossmatches to linearly increase the number of lives saved,[4] and moreover our results suggest that practical optimization approaches are available.

REFERENCES

ABRAHAM, D. J., BLUM, A., AND SANDHOLM, T. 2007. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proc. of 8th EC*. 295–304.

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

ASHLAGI, I., FISCHER, F., KASH, I., AND PROCACCIA, A. D. 2010. Mix and match. In *Proc. of 11th EC*. 305–314.

ASHLAGI, I., GAMARNIK, D., REES, M. A., AND ROTH, A. E. 2012. The need for (long) chains in kidney exchange. NBER Working Paper Series No. 18202.

ASHLAGI, I. AND ROTH, A. 2011. Individual rationality and participation in large scale, multi-hospital kidney exchange. In *Proc. of 13th EC*. 321–322.

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.

CARAGIANNIS, I., FILOS-RATSIKAS, A., AND PROCACCIA, A. D. 2011. An improved 2-agent kidney exchange mechanism. In *Proc. of 7th WINE*. 37–48.

CHEN, N., IMMORLICA, N., KARLIN, A. R., MAHDIAN, M., AND RUDRA, A. 2009. Approximating matches made in heaven. In *Proc. of 36th ICALP*. 266–278.

COSTELLO, K. P., TETALI, P., AND TRIPATHI, P. 2012. Matching with commitment. In *Proc. of 39th ICALP*. 822–833.

DICKERSON, J. P., PROCACCIA, A. D., AND SANDHOLM, T. 2012a. Dynamic matching via weighted myopia with application to kidney exchange. In *Proc. of 26th AAAI*. 1340–1346.

DICKERSON, J. P., PROCACCIA, A. D., AND SANDHOLM, T. 2012b. Optimizing kidney exchange with transplant chains: Theory and reality. In *Proc. of 11th AAMAS*. 711–718.

MOLINARO, M. AND RAVI, R. 2011. The query-commit problem. CoRR abs/1110.0990.

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*, 828–851.

TOULIS, P. AND PARKES, D. C. 2011. A random graph model of kidney exchanges: efficiency, individual-rationality and incentives. In *Proc. of 12th EC*. 323–332.

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

WALKUP, D. W. 1980. Matchings in random regular bipartite digraphs. *Discrete Mathematics*.

ZENIOS, S., WOODLE, E. S., AND ROSS, L. F. 2001. Primum non nocere: Avoiding harm to vulnerable candidates in an indirect kidney exchange. *Transplantation 72*, 648–654.

[4] This statement would be trivial to formally establish for, e.g., complete graphs.

Online Appendix to: Harnessing the Power of Two Crossmatches

AVRIM BLUM, Carnegie Mellon University
ANUPAM GUPTA, Carnegie Mellon University
ARIEL D. PROCACCIA, Carnegie Mellon University
ANKIT SHARMA, Carnegie Mellon University

A. PROOF OF LEMMA 6.6

Consider the optimal subgraph $H(V, E')$. If H does not already satisfy the stated property, we show how to convert it into one that satisfies the stated property and does not reduce the expected size of its maximum matching.

We can assume that subgraph H satisfies the properties stated in Corollary 5.2. We will now present the procedure to convert H into one that satisfies the properties stated in the statement of the theorem.

```
(1) Let $S \leftarrow [k]$.
(2) While $S$ is non-empty
    — Pick a $j \in S$, such that there exists a vertex $u \in Q_j$ with no edges incident to it
      under $E'$.
    — If for some $v \in P_j$ either of $b$ or $d$ are not members of $Q_j$, say it is $b$, we shall
      replace edge $(v, b)$ by $(v, u)$ in $E'$.
    — If there does not exist a $v \in Q_j$ such that $v$ has an edge incident to a vertex not
      in $Q_j$, remove $j$ from $S$.
```

First we show that the above procedure is well-defined and that it terminates.

CLAIM A.1. *(Well-defined) In each iteration of the while loop, in the first step of the loop, we can find a $j \in S$ and a vertex $u \in Q_j$ such that no edges are incident to it under E'.*

PROOF. Since

(1) the total number of edges in E' that are incident to the vertices in the set $\cup_{i \in S} P_i$ can be at most $2 \cdot \sum_{i \in S} |P_i|$ ($\because \forall v \in V$, $\delta_{E'}(v) \leq 2$), and
(2) E, and therefore E', does not contain any edge going between a vertex in R and a vertex in $\cup_{i=1}^{k} Q_i$ or an edge going between a vertex in Q_i and a vertex in Q_j for any $1 \leq i, j \leq k$,

hence the number of edges incident to vertices in $\cup_{i=1}^{k} Q_j$ under edge set E' is at most $2 \cdot \sum_{i \in S} |P_i|$. On the other hand, the cardinality of the set $\cup_{i=1}^{k} Q_j$ is strictly greater than $2 \cdot \sum_{i \in S} |P_i|$. Hence, there must exist a vertex $u \in Q_j$ for some $j \in S$, such that u does not have any edge incident to it. □

CLAIM A.2. *(Loop terminates) The while loop eventually terminates.*

PROOF. After each iteration of the while loop, the number $|E' \cap \bigcup_{i=1}^{k} (P_i \times Q_i)|$ increases by one. And this number is upper bounded by $2 \cdot \sum_{i=1}^{k} |P_i|$. □

The following claim states that the subgraph $H(V, E')$ satisfies the properties stated in Corollary 5.2 at all points of the execution of the procedure.

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

CLAIM A.3. *At all points in the execution of the procedure (including the point when it terminates), the subgraph $H(V, E')$ satisfies the properties stated in Corollary 5.2.*

PROOF. Before the start of the procedure, we had assumed that the subgraph H satisfies the properties stated in Corollary 5.2, and at no step in the above procedure, we make a move that can violate the properties stated in Corollary 5.2. □

We now show that the expected size of matching does not change at any step of the procedure.

CLAIM A.4. *(No change in solution quality) In each iteration of the while loop, the change made to E' in the second step of the loop, does not change the expected size of matching*

PROOF. In any iteration, the pair (j, u) found in the first step of the iteration satisfy the property u has an edge to each vertex $v \in P_j$ in E. Hence by Corollary 5.2 and Claim A.3, each $v \in P_j$ must be incident to two nodes b and d, such that $\delta_{E'}(b) = 1 = \delta_{E'}(d)$.

The second step changes E' only if there exists a $v \in P_j$ that has its edges incident to a b and d, such that at least one of b or d is outside Q_j. Suppose b is the vertex outside Q_j. In such a case, edge (v, b) is replaced with (v, u).

Since before replacement vertex b has only one edge incident to it and that was to v, by replacing edge (v, b) by (v, u) in E', from the point of view of matching, we have only switched the situation of b and u, and left the situation of v unchanged, and so the expected maximum matching size does not change. □

The final claim shows that the procedures converts H into one that satisfies the properties stated in the statement of the theorem.

CLAIM A.5. *At the end of the procedure, the subgraph $H(V, E)$ has the property that for every $1 \leq i \leq k$, all vertices $v \in P_i$ have, in E', two edges incident to a unique pair of vertices in $Q_i \times Q_i$. No other edges are present in E'.*

PROOF. The procedure terminates when the set S becomes empty. Since $S = [k]$ at the beginning of the procedure, hence, for all elements $j \in [k]$, there is a point during the execution when j is removed from set S.

For any element j, consider the point it is removed from set S. That can occur only under the circumstance that all edges that are incident to vertices in P_j have their other ends in Q_j. Furthermore, since the subgraph H at all points in the execution of the procedure, satisfies the properties in Corollary 5.2, hence we have the property that all vertices $v \in P_i$ have, in E', two edges incident to a unique pair of vertices in $Q_i \times Q_i$.

Hence, by end of the procedure, we have the property that for every $1 \leq i \leq k$, all vertices $v \in P_i$ have, in E', two edges incident to a unique pair of vertices in $Q_i \times Q_i$. These edges exhaust the total number of edges that could have been incident to the set $\cup_{i=1}^{k} P_i$ since each vertex can have at most two edges incident to it in E'. Furthermore, note that the graph G does not contain any edges in the set $Q_i \times Q_j$ for any $1 \leq i, j \leq k$. Hence, no other edges can occur in H other than those already listed. □

This completes the proof of the lemma. □

B. PROOF OF LEMMA 5.1

If $\delta_{E'}(v) < 2$, then we can add the edge (u, v) without removing any edge. Adding an edge cannot decrease the expected size of matching.

Hence, let us consider the case where $\delta_{E'}(v) = 2$, and say, the vertices to which v has an edge are b and d. If either d or b, has exactly one edge incident to it in E', say it is

d, then we can drop the edge (v, d) in E' and add the edge (v, u). This does not change the expected size of matching since up to renaming, nothing has changed in the graph (u and d have exactly the same set of characteristics).

Therefore, we are left with the case where $\delta_{E'}(d) = 2 = \delta_{E'}(b)$. Consider E'', which the same as E' except that we drop the edge (v, d) and add the edge (v, u). We would like to show that the expected size of matching in E'' is at least as much as in E'.

In order to show that the expected size of matching in E'' is at least as much as in E', we shall partition the sample space of outcomes as follows:

(1) Both (v, d) and (v, u) are present: Consider any outcome ω of edges in E where both (v, d) and (v, u) exist. Consider a maximum matching M for E' in ω. If M matches v to b, then M is also *a matching* in E'' since $E' \Delta E'' = \{(v, u), (v, d)\}$ and both don't exist in M. Hence the cardinality of the maximum matching in E'' is at least $|M|$. If M matches v to d, then consider matching M' for E'' which is exactly the same as M, but that it matches v to u (and not v to d). Again, $|M'| = |M|$, and hence the cardinality of the maximum matching in E'' is at least $|M'| = |M|$.

(2) Both (v, d) and (v, u) are absent: Consider any outcome ω of edges in E where both (v, d) and (v, u) exist. Consider a maximum matching M for E' in ω. M is also *a matching* in E'' since $E' \Delta E'' = \{(v, u), (v, d)\}$ and both don't exist in M. Hence the cardinality of the maximum matching in E'' is at least $|M|$.

(3) Exactly one of (v, d) and (v, u) is present: Clearly for any outcomes of edges, the size of maximum matching in E' and E'' can differ by at most one. For a particular outcome of edges ω, denote the size of maximum matching for E' and E'' by $\phi(E', \omega)$ and $\phi(E'', \omega)$ respectively.

We shall partition the sub-sample space (where exactly one of the two edges is present) into two halves. In one half, edge (v, d) would be present and (v, u) absent, and in the other half, the opposite would be true. Furthermore, we shall have a one-to-one mapping from points in the first half to that in the second half. The two points that are mapped to each other shall carry the same probability. In addition, we shall have the property that if for a particular point ω in say, the first half, $\phi(E', \omega) - \phi(E'', \omega) = 1$, then for the sample point ω' in the other half that ω maps to, $\phi(E', \omega') - \phi(E'', \omega') = -1$. Hence, in expectation over this sub-sample space, the size of matching matching of E' will be no more than that of E''.

We now show the construction of the two halves and the mapping between them. Fix the outcome ω' of all edges in E but for (v, d) and (v, u). Let ω_u be ω' with (v, u) present and (v, d) absent, and let ω_d be ω' with (v, d) present and (v, u) absent. Consider the set A of points ω_d that we generate while enumerating over ω'. Similarly, consider the set B, that consists of points ω_u, again while enumerating over all possible ω'. It is easy to see that A and B are disjoint, and that their union captures the sub-sample space where exactly one of (v, d) and (v, u) is present. Also, again it is easy to verify that $|A| = |B|$. A and B shall constitute our two halves.

We now explain the mapping from A to B. It will be the natural mapping, where $\omega_1 \in A$ and $\omega_2 \in B$ are mapped to each other, in case the outcome of all edges but for (v, d) and (v, u) is the same in ω_1 and ω_2. It is easy to see that this is a well defined one-to-one map and that both points that are mapped to each other carry the same probability weight.

Consider a $\omega_d \in A$ and $\omega_u \in B$ that are mapped to each other. Consider the set S of all maximum matchings for E' in outcome space ω_d.

(a) Either there exists a maximum matching M in set S that does not use the edge (v, d).

In this case, $\phi(E'', \omega_d) \geq \phi(E', \omega_d) = |M|$ because M is also *a matching* in E'' since it does not use the edge (v, d) which is the only edge in $E'' \setminus E'$.

Moreover, $\phi(E'', \omega_u) \geq \phi(E', \omega_u)$ since under outcome ω_u, edge (v, d) is absent and hence, the maximum matching M for E' under ω_u shall also be a matching in E'' under ω_u.

(b) Or every maximum matching in S uses the edge (v, d).
As we have claimed earlier, that since $|E'' \backslash E'| = 1$, hence $\phi(E', \omega_d) - \phi(E'', \omega_d) \leq 1$.
Moreover, we have that $\phi(E', \omega_u) - \phi(E'', \omega_u) \leq -1$. Why is this the case? Well, consider any maximum matching M for E' under ω_d. We know that M uses edge (v, d). Construct M' which has all the edges as in M but has edge (v, u) replacing (v, d). M' is a legal matching for E'' under ω_u. Hence, $\phi(E'', \omega_u) \geq |M'| = |M| = \phi(E', \omega_d)$. Moreover, it is the case that $\phi(E', \omega_u) \leq \phi(E', \omega_d)$ for under ω_d, E' has strictly a superset of edges present as compared to in ω_u. Not only that, it is also the case that $\phi(E', \omega_u) \leq |M| - 1$, for if it were the case that $\phi(E', \omega_u) = |M|$, then it means that there exists a matching in E' that has cardinality equal to $|M|$ and does not use the edge (v, d) contradicting the assumption of this subcase that every maximum matching in S uses the edge (v, d).
In summary, for this subcase we have, $\phi(E', \omega_d) - \phi(E'', \omega_d) \leq 1$ and $\phi(E', \omega_u) - \phi(E'', \omega_u) \leq -1$.

Hence, in each one of the above cases, we have that in expectation over the sub-sample space considered in the case, $\phi(E', \omega) - \phi(E'', \omega) \leq 0$. And hence, in expectation over all of sample space, $\phi(E', \omega) - \phi(E'', \omega) \leq 0$. □

C. DISTRIBUTION ON GRAPHS

Assume that for a particular graph $G(V, E)$ we have been able to characterize an optimal subgraph $H(V, E')$. Moreover, we can find the subgraph H algorithmically. However, what if we are not dealing with G, but rather a graph G_r which has the same vertex set as G and whose edges are drawn from the following distribution: every edge $e \in E$ is included in G_r with some constant probability c. Can we somehow use the fact that we have been able to solve the problem for G, and use its solution for G_r?

We would like to emphasize here that the aim of this section is to solve DOUBLE-CROSSMATCH problem for *the graph* G_r. In solving it, we would like to leverage the fact that we know that it is drawn from the underlying graph G and have the knowledge of an optimal or almost optimal solution for DOUBLE-CROSSMATCH problem for G.

OBSERVATION C.1. *The expected matching size of an optimal solution for the complete graph G, denoted by $opt(G)$, is at least as much as the expected matching size of an optimal solution for any graph G_r, denoted by $opt(G_R)$.*

The reason for the above observation is that the edge set of G_r is a subset of the edge set of G, and hence any solution for G_r, i.e., a subgraph H_r of G_r, is also a subgraph of G. Hence, $opt(G) \geq opt(G_r)$. One corollary of the above observation is the following.

COROLLARY C.2. *If for a graph G_r, drawn from G, we can algorithmically find a subgraph H_r, that has expected matching size withing some additive loss of $opt(G)$, then that implies that the expected matching size of H_r is at least $opt(G_r)$ within the same additive loss.*

Hence, if we wish to prove that a particular subgraph H_r for a graph G_r has expected matching size close to $opt(G_r)$, it suffices to show that the expected matching size of H_r is close to $opt(G)$. In this section, we explore this question for various special cases of G.

Before we delve into the special cases, we would like to state a result on random bipartite graph.

CLAIM C.3. *Consider a complete bipartite graph $G(P \cup Q, P \times Q)$. Draw a random bipartite graph $G_r(P \cup Q, E')$, where every edge in $P \times Q$ is included in E independently with probability c, for some constant c. There exists n_0 (a constant depending on c) such that if $n = \min(|P|, |Q|) \geq n_0$, then with probability at least $1 - o(\frac{1}{n})$, there exists a bipartite matching in G_r of size n.*

PROOF. Consider the case when $|P| \leq |Q|$; the other case can be taken care of similarly. Let $|P| = n$, and let us consider an arbitrary subset $Q' \subseteq Q$, such that $|Q'| = n$. We shall show that the random bipartite graph $G'_r(P \cup Q', E' \cap (P \times Q'))$ has a perfect matching with probability at least $1 - o(\frac{1}{n})$.

We first show that with probability at least $1 - \frac{1}{n^2}$, every vertex in both sets P and Q' has degree at least 3 in graph G'_r. Consider a particular vertex $v \in P \cup Q'$. Consider the n independent random variables, each taking value in $\{0, 1\}$ and representing a possible edge between v and a vertex from the opposite side. Let these random variable be $X_1, \cdots, X_n$. Since each X_i takes value 1 with probability c, hence the expected degree of vertex v, $\mathbb{E}[\sum_{i=1}^{n} X_i] = cn$.

Let $n_0 \geq 6/c$. If $\sum_{i=1}^{n} X_i \leq 3$ (i.e., vertex v has degree at most 3), then in particular, $\sum_{i=1}^{n} X_i \leq \frac{cn}{2}$. By Chernoff bound, $\Pr[\sum_{i=1}^{n} X_i \leq \frac{cn}{2}] \leq \exp(-cn/8)$.

By union bound, the probability that at least one vertex in $P \cup Q'$ has degree less than 3 in graph G'_r is at most $2n \cdot \exp(-cn/8)$. Let n_0 be the minimum integer $\geq 6/c$, such that $2n_0 \cdot \exp(-cn_0/8) \leq \frac{1}{n_0^2}$. We then have that for all $n \geq n_0$, with probability at least $1 - \frac{1}{n^2}$, every vertex in both sets P and Q' has degree at least 3 in graph G'_r.

Let us condition the analysis from here on to each vertex in $P \cup Q'$ having degree at least 3 in graph G'_r. Clearly, once we condition, then each vertex in graph G'_r has at least three *random* neighbors from the opposite side. We can now apply Walkup's theorem [Walkup 1980] to conclude that there exists a perfect matching in G'_r with probability at least $1 - o(\frac{1}{n})$.

Removing the conditioning, we get that with probability at least $(1 - o(\frac{1}{n})) \cdot (1 - \frac{1}{n^2}) = 1 - o(\frac{1}{n})$, there exists a perfect matching in G'_r, and hence a matching of size n in G_r. □

Remark C.4. For all the graphs that we consider below, we shall assume that the number of vertices in the graph is large enough to apply Claim C.3.

C.1. Complete Graph

Suppose that G is a complete graph. If $|V|$ is divisible by 4, then using Corollary 3.5, we know that the optimal subgraph H for G is a cover of the vertices of G through 4-cycles. We now use this result for the complete graph G to get the polynomial time Algorithm 2, which with high probability, gives an almost optimal solution for G_r.

LEMMA C.5. *Algorithm 2, in polynomial time, constructs a subgraph H_r whose expected size of matching, with probability at least $1 - o(\frac{1}{|V|})$, over the draw of random graph G_r from a complete graph G, is at least $opt(G) - O(1) \geq opt(G_r) - O(1)$.*

We first prove an important claim.

CLAIM C.6. *Over the draw of G_r, with probability at least $1 - o(\frac{1}{|V|})$, the subgraph H_r computed using Algorithm 2 has $|V|/4 - O(1)$ vertex disjoint 4-cycles.*

PROOF. Having thrown out $O(1)$ vertices in Step 1, consider any fixed partition (A, B) of the remaining vertices for Step 2 of Algorithm 2, with $|A| = |B|$ and fixed pairing up of vertices in A and in B to get A' and B'. Consider a particular pair of vertices $v_{xy} \in A$ and $v_{st} \in B$. Over the draw of G_r, what is the probability that an edge exists between v_{xy} and v_{uv}? For an edge to exist between these two vertices, the edges $(x, s), (s, y), (y, t), (t, x)$ must exist in G_r. Each of these edge exists independently with probability c in G_r, and hence all four exist with probability c^4.

Therefore, between any pair of vertices $v_{xy} \in A$ and $v_{st} \in B$, an edge exists with probability c^4. Using Claim C.3 and Remark C.4, with probability at least $1 - o(\frac{1}{|V|})$ over the draw of G_r, the bipartite graph between A and B admits a maximum matching of size $|A| = |B|$. This in turn implies that with probability at least $1 - o(\frac{1}{|V|})$, the subgraph H_r constructed for G_r has at least $|V|/4 - O(1)$ vertex disjoint 4-cycles. □

PROOF OF LEMMA C.5. It is easy to see that Corollary 3.5 implies that the subgraph H for G with $\lfloor |V|/4 \rfloor$ vertex-disjoint 4-cycles has expected size at least $opt(G) - O(1)$ where we lose $O(1)$ if the cardinality of the vertex set of G is not divisible by 4. The expected size of maximum matching in H is $\lfloor |V|/4 \rfloor \cdot M_{4C}$, where M_{4C} is the expected size of maximum matching in a single 4-cycle. In other words, $opt(G) \leq \lfloor |V|/4 \rfloor \cdot M_{4C} + O(1)$.

Claim C.6 shows that Algorithm 2 produces a subgraph H_r that with probability at least $1 - o(\frac{1}{|V|})$, has at least $|V|/4 - O(1)$ vertex disjoint 4-cycles. Hence, the with probability at least $1 - o(\frac{1}{|V|})$, the expected size of maximum matching in H_r is at least $opt(G) - O(1)$. Moreover, from Observation C.1, $opt(G) \geq opt(G_r)$. Hence the result. □

C.2. Almost balanced bipartite graphs

We now consider G that is a complete bipartite graph between the two sets of vertices L and R with $|L| \leq |R| \leq 2|L|$. From Lemma 4.1, we know that an optimal subgraph H for the graph G consists of 4-cycles and paths of length 2 (plus maybe a path of length 4 or an edge). Furthermore, by Lemma 4.1, it is easy to discern that a subgraph H which has $\lfloor x \rfloor$ 4-cycles and y paths of length 2 has expected matching size at least $opt(G) - O(1)$ where x and y are given by $2 \cdot x + y = |L|$ and $2 \cdot x + 2 \cdot y = |R|$. We now utilize this knowledge to build Algorithm 4 for construct a subgraph H_r for a randomly drawn graph G_r from G. The guarantee of this algorithm can be easily inferred from the preceding discussion and the following claim.

CLAIM C.7. *Given bipartite graph $G(L \cup R, L \times R)$ with $|L| \leq |R| \leq 2|L|$, Algorithm 4, in polynomial time, constructs a subgraph H_r of G_r, that, with probability at least $1 - o(\frac{1}{T})$, over the draw of G_r, has $x - o(T)$ 4-cycles and $y - o(T)$ paths of length 2, where x and y are given by $2 \cdot x + y = |L|$ and $2 \cdot x + 2 \cdot y = |R|$, and $T = |L \cup R|$.*

PROOF. If both x and y are $\Omega(T)$, we can apply Claim C.3 to each of the bipartite matchings M_1 and M_2, found in Step 5 of Algorithm 4, to infer that

(1) with probability at least $1 - o(\frac{1}{x})$, M_1 has size $\lfloor x \rfloor$, and hence, the number of 4-cycles in H_r is $\lfloor x \rfloor$
(2) with probability at least $1 - o(\frac{1}{y})$, M_2 has size y, and hence, the number of paths of length 2 in H_r is y

where H_r is the subgraph returned by Algorithm 4. Hence, we can infer that with probability at least $1 - o(\frac{1}{T})$, H_r has $\lfloor x \rfloor$ 4-cycles and y paths of length 2.

On the other hand, if one of x or y is $o(T)$, then we can ignore the contribution from that term, and applying Claim C.3 solely to the other term, get that with probability at least $1 - o(\frac{1}{T})$, H_r has $x - o(T)$ 4-cycles and $y - o(T)$ paths of length 2. □

C.3. Lopsided bipartite graphs

Let G be a complete bipartite graph between the two sets L and R, but with $|L| < \frac{1}{2}|R|$. By Lemma 6.6, we know that the optimal subgraph H for G has each vertex in L having an edge each to distinct and unique vertices in R, and this implies a total of $|L|$ vertex disjoint paths of length 2 in H. Hence, $opt(G) = |L| \cdot M_{2P}$, where M_{2P} is the expected size of maximum matching in a path of length 2. We build Algorithm 3 for such a bipartite graph, and the guarantee of the algorithm can be easily inferred from the following claim.

CLAIM C.8. *Given a graph $G(L \cup R, L \times R)$, with $|L| < \frac{1}{2}|R|$, Algorithm 3, in polynomial time, constructs a subgraph H_r, which has expected matching size, with probability at least $1 - o(\frac{1}{|A|})$, over the draw of the graph G_r, has $|L|$ paths of length 2.*

PROOF. Applying Claim C.3 to the matching found in Step 3, we can see that with probability at least $1 - o(\frac{1}{|A|})$, we find a perfect bipartite matching and hence the subgraph H_r returned by the algorithm has $|L|$ vertex disjoint paths of length 2. Hence the claim follows. □

D. PROOF OF CLAIM 4.3

We enumerate here the proofs of the various parts of the claim.

(1) The proof of Lemma 3.2 shows not only 4-cycle has the highest expected number of matched edges, but that among even length cycles, a node has strictly higher expected probability of being matched in cycles of length l than in $l + 2$ for all even $l \geq 4$.

(2) The expected number of matched edges in a 4-cycle is $2p^2 + 2p(1-p)^2 + 2p(1-p)(1+p^2)$, for an edge is $1 - (1-p)^2$ and for a cycle of length 6 is $6p(1-p)^2(1 + p/2 + p^2 + p^3/2 + p^4) + 3p^2$. We can verify that the sum of the first two expressions strictly dominates the third for all $p \in (0, 1)$.

(3) For a 4-cycle, the expected number of matched edges is $2p^2+2p(1-p)^2+2p(1-p)(1+p^2)$; for a path of length 2, the expected number of matched edges is $1 - (1-p)^2$ and for the case of a path of length 4, the expected number of matched edges is $2p^2 + 2p(1-p)^2 + p(1-p)^2 + p(1-p)(1+p^2)$.
Summing the first expression with twice the second, and simplifying shows that it is equal to twice the third.

□