The Stochastic Matching Problem with (Very) Few Queries

SEPEHR ASSADI, University of Pennsylvania
SANJEEV KHANNA, University of Pennsylvania
YANG LI, University of Pennsylvania

Motivated by an application in *kidney exchange*, we study the following *stochastic matching* problem: we are given a graph $G(V, E)$ (not necessarily bipartite), where each edge in E is *realized* with some constant probability $p > 0$ and the goal is to find a maximum matching in the realized graph. An algorithm in this setting is allowed to make *queries* to edges in E in order to determine whether or not they are realized.

We design an *adaptive* algorithm for this problem that, for any graph G, computes a $(1-\varepsilon)$-approximate maximum matching in the realized graph G_p with high probability, while making $O\Big(\frac{\log(1/\varepsilon p)}{\varepsilon p}\Big)$ queries per vertex, where the edges to query are chosen adaptively in $O\Big(\frac{\log(1/\varepsilon p)}{\varepsilon p}\Big)$ rounds. We further present a *non-adaptive* algorithm that makes $O\Big(\frac{\log(1/\varepsilon p)}{\varepsilon p}\Big)$ queries per vertex and computes a $(\frac{1}{2}-\varepsilon)$-approximate maximum matching in G_p with high probability.

Both our adaptive and non-adaptive algorithms achieve the same approximation factor as the previous best algorithms of Blum *et al.* (EC 2015) for this problem, while requiring *exponentially smaller* number of per-vertex queries (and rounds of adaptive queries for the adaptive algorithm). Our results settle an open problem raised by Blum *et al.* by achieving only a polynomial dependency on both ε and p. Moreover, the approximation guarantee of our algorithms is *instance-wise* as opposed to only being competitive in *expectation* as is the case for Blum *et al.* . This is of particular relevance to the key application of stochastic matching in kidney exchange. We obtain our results via two main techniques, namely *matching-covers* and *vertex sparsification* that may be of independent interest.

General Terms: Algorithms, Economics

Additional Key Words and Phrases: Stochastic matching, Kidney exchange

1. INTRODUCTION

We study the problem of finding a maximum matching in presence of *uncertainty* in the input graph. Specifically, we consider the *stochastic matching* problem in which we are given an undirected graph $G(V, E)$ where each edge $e \in E$ is *realized* with some constant probability $p > 0$ and the goal is to find a maximum matching in the realized graph. To find a large matching, an algorithm is allowed to *query* edges in E to determine whether or not they are realized.

There is a trivial solution for the stochastic matching problem: simply query all edges in E and compute a maximum matching over the realized graph. However, in many applications, determining whether or not an edge is realized could be both costly and time consuming (we will elaborate more on this in the next section). Consequently, to minimize cost, it is preferable that an algorithm queries as few edges as possible, and to minimize the time consumed in the query process, an algorithm should have

Authors' addresses: Department of Computer and Information Science, University of Pennsylvania. Email: {sassadi,sanjeev,yangli2}@cis.upenn.edu.

EC'16, July 24–28, 2016, Maastricht, The Netherlands. ACM 978-1-4503-3936-0/16/07 ...$15.00.

http://dx.doi.org/10.1145/2940716.2940769

only a few rounds of *adaptivity* whenever possible (or minimize the *degree of adaptivity*), meaning that it is preferable if the decision of "which edges to query next" does not depend on the outcome of previous queries since in this case, many edges can be queried in parallel. The formal definition of this model is presented in Section 1.2.

1.1. Kidney Exchange

A canonical and arguably the most important application of the stochastic matching problem appears in *kidney exchange*. Typically, organ donation comes from deceased donors since many organs cannot be harvested from a living donor without jeopardizing the health of the donor. Fortunately, kidney is an exception, and a healthy living donor is able to donate one of his/her two kidneys without facing major life threatening consequences.

The possibility of having living donors triggers the idea of kidney exchange: often patients have a family member who is willing to donate his/her kidney, but this kidney might not be a suitable match for the patient due to reasons like incompatible blood-type etc. To solve this problem, a kidney exchange is performed in which patients *swap* their incompatible donors to each get a compatible donor.

This setting can be modeled as a maximum matching problem as follows. Create a graph $G(V, E)$ where each patient-donor pair is a vertex and there is an edge between two vertices iff the two patient-donor pairs can perform a kidney exchange. Consequently, a maximum matching in this graph identifies the maximum number of patient-donor pairs that are able to perform an exchange.

To construct such a graph for kidney exchange, typically we only have access to the medical record of the patients and the donors, which contains information like blood type, tissue type, etc. This information can be used to rule out the patient-donor pairs where donation is impossible (e.g., different blood types), but does not provide a conclusive answer for whether or not a donation is indeed feasible. In order to be (more) certain that a donor can donate to a patient, more accurate tests must be performed before the transplant, that includes crossmatching, antibody screen, etc[1], which are both costly and time consuming.

The stochastic matching problem captures the essence of the need of extra tests for kidney exchange: an algorithm selects a set of patient-donor pairs to perform the extra costly and time consuming tests (i.e., query the edges), while making sure that w.h.p. there is a large matching among the pairs that pass the extra tests (i.e., in the realized graph). The objective of querying as few edges as possible captures the essence of minimizing the total cost and the objective of having small degree of adaptivity captures the essence of minimizing patient's waiting time by performing the extra exams between many patient-donor pairs in parallel.

The kidney exchange problem has been extensively studied in the literature, particularly under stochastic settings (see, e.g., [Akbarpour et al. 2014; Anderson et al. 2015a,b; Awasthi and Sandholm 2009; Dickerson et al. 2012, 2013; Dickerson and Sandholm 2015; Manlove and O'Malley 2014; Unver 2010]); we refer the interested reader to [Blum et al. 2015] for a more detailed discussion.

1.2. Model

We now formally define the model for the stochastic matching problem. For any graph $G(V, E)$, let $\textsc{opt}(G)$ denote the *maximum matching size* in G. With a slight abuse of notation, we sometimes also use $\textsc{opt}(X) := \textsc{opt}(G(V, X))$ for $X \subseteq E$, i.e., X is a set of edges instead of a graph. Throughout, we use n to denote $|V|$.

[1] American Transplant Foundation, http://www.americantransplantfoundation.org/.

In the stochastic setting, for the input graph $G(V, E)$, each edge in E is realized *independently* w.p.[2] p. When sampling each edge w.p. p, for any set of edges $X \subseteq E$, we slightly abuse the notation and use X_p to denote both a random variable that corresponds to sampling edges in X w.p. p, as well as a specific realization of the random variable X_p. We call each possible realized graph $G(V, E_p)$ a *realization* of G.

In the *stochastic matching problem*, we are given a graph $G(V, E)$ and our goal is to compute a matching M in $G(V, E_p)$, such that w.h.p. (taken over both the randomness of the algorithm and the randomness of the realization $E_p \subseteq E$), the size of M is close to $\text{OPT}(E_p)$. An algorithm is allowed to *query* any edge $e \in E$ to determine whether or not $e \in E_p$, and we consider the following two classes of algorithms.

— **Non-adaptive algorithms.** A non-adaptive algorithm specifies a subset of edges $Q \subseteq E$, queries all edges in Q in parallel, and outputs a matching among the edges realized in Q.
— **Adaptive algorithms.** An adaptive algorithm proceeds in *rounds* where in each round, based on the edges queried and realized thus far, the algorithm chooses a new set of edges to query in parallel. We say the *degree of adaptivity* of an algorithm is d if the algorithm makes at most d rounds of adaptive queries.

In general, the goal is to design algorithms where the number of per-vertex queries is independent of n. For adaptive algorithms, the degree of adaptivity is further required to be independent of n.

We remark that throughout, we always assume $\text{OPT}(G) = \omega(1/p)$ to obtain the desired concentration bounds. This assumption essentially says that the expected matching size in the realized graph is bounded from below by a sufficiently large constant.

1.3. Related Work

Prior to our work, the state of the art adaptive and non-adaptive algorithms for stochastic matching are that of [Blum et al. 2015], which is an adaptive (resp. non-adaptive) algorithm that achieves a $(1 - \varepsilon)$-approximate (resp. $(\frac{1}{2} - \varepsilon)$-approximate) matching *in expectation*, while both the number of per-vertex queries and the degree of adaptivity (for the adaptive algorithm) is $\frac{1}{p^{O(1/\varepsilon)}}$. Note that while in these algorithms, the number of per-vertex queries and degree of adaptivity is independent of n, the exponential dependence on $\frac{1}{\varepsilon}$, limits the practical appeal of the algorithm. [Blum et al. 2015] raised an open problem regarding the possibility of avoiding the exponential dependency on $\frac{1}{\varepsilon}$ for both the number of per-vertex queries and the degree of adaptivity.

Other variants of the stochastic matching setting have also been studied in the literature. [Blum et al. 2013] considered the setting where each vertex can only pick two incident edges to query and the goal is to find the optimal set of edges to query. Another well studied setting is the *query-commit* model, whereby if an algorithm decides to query an edge e, then e must be part of the matching the algorithm outputs in case e is realized [Adamczyk 2011; Bansal et al. 2012; Chen et al. 2009; Costello et al. 2012; Gupta and Nagarajan 2013].

1.4. Our Results

We provide algorithms for the stochastic matching problem with exponentially smaller number of queries and degree of adaptivity (for the adaptive algorithm) compared to the best previous bounds of [Blum et al. 2015]. In particular,

[2]Throughout, we use *w.p.* and *w.h.p.* to abbreviate "with probability" and "with high probability", respectively.

THEOREM 1.1 (INFORMAL). *For any $\varepsilon > 0$, there exists a poly-time adaptive $(1-\varepsilon)$-approximation algorithm for the stochastic matching problem which queries $O\left(\frac{\log(1/\varepsilon p)}{\varepsilon p}\right)$ edges per vertex and has degree $O\left(\frac{\log(1/\varepsilon p)}{\varepsilon p}\right)$ of adaptivity.*

The formal statement of Theorem 1.1 is presented in Section 4 (as Theorem 4.1).

THEOREM 1.2 (INFORMAL). *For any $\varepsilon > 0$, there exists a poly-time non-adaptive $(\frac{1}{2}-\varepsilon)$-approximation algorithm for the stochastic matching problem which queries $O\left(\frac{\log(1/\varepsilon p)}{\varepsilon p}\right)$ edges per vertex.*

The formal statement of Theorem 1.2 is presented in Section 5 (as Theorem 5.1).

These results provide an affirmative answer to an open question raised by [Blum et al. 2015] regarding the possibility of avoiding the exponential dependency on $1/\varepsilon$ for both the number of per-vertex queries and the degree of adaptivity.

One of the key property of our results is that we provide *instance-wise* approximation guarantees, i.e., for $1-o(1)$ fraction of the realizations of G, the algorithm outputs a competitive solution. This is a *stronger* guarantee than the *expectation* guarantee provided in [Blum et al. 2015] which states that expected size of the matching output by the algorithm is competitive with the expected size of the maximum matching among all realizations. We remark that our instance-wise guarantee is of particular interest to the key application of the kidney exchange problem.

Finally, it is worth mentioning that even when the input graph is a complete graph, one needs to query $\Omega(\log(1/\varepsilon)/p)$ edges per each vertex to simply ensure that the number of isolated vertices in the realized graph is at most εn. This is due to the fact that if one queries less than $\ln(1/\varepsilon)/2p$ edges per vertex, the probability that no edge incident on a vertex is realized is $(1-p)^{\ln(1/\varepsilon)/2p} \geq \exp(-2p \cdot \ln(1/\varepsilon)/2p) = \varepsilon$. On the other hand, it is easy to see that for any constant $p > 0$, any realization of a complete graph has a perfect matching w.h.p. Hence, $\Omega(\log(1/\varepsilon)/p)$ is a simple lower bound on the number of per-vertex queries for any $(1-\varepsilon)$-approximation algorithm even on complete graphs. Our per-vertex query bounds in Theorem 1.1 and Theorem 1.2 only ask for slightly more than this simple lower bound.

1.5. Our Techniques

To explain the high-level idea underlying our algorithms, it will be convenient to focus on the case when G_p has a perfect matching; however, we emphasize that our algorithms do not require this property. The idea behind both of our algorithms is to construct a *matching-cover* of the input graph G and query the edges of the cover in the algorithm. Roughly speaking, a γ-matching-cover of a graph $G(V,E)$ is a collection of matchings of G of size $\gamma \cdot (|V|/2)$ that are essentially edge-disjoint (see Section 3.1 for a formal definition). One of the main technical ingredient of our work is a structural result proving that: for any algorithm that outputs a γ-matching cover with $\Theta(1/\varepsilon p)$ matchings, w.h.p. the set of realized edges in the cover contains a matching of size $(1-\varepsilon) \cdot \gamma \cdot (|V|/2)$. We prove this result through a constructive argument based on the Tutte-Berge formula (see Section 2 for more detail on Tutte-Berge formula).

Next, we show that there is a simple adaptive (resp. non-adaptive) algorithm that computes a 1-matching-cover (resp. $1/2$-matching-cover) with $\Theta(1/\varepsilon p)$ matchings, which immediately implies that w.h.p. the algorithm achieves a $(1-\varepsilon)$-approximation (resp. $(\frac{1}{2}-\varepsilon)$-approximation).

Finally, to eliminate the assumption that G_p has a perfect matching, we establish a *vertex sparsification lemma* (see Section 3.2) which allows us to reduce the number of vertices in any instance G from $|V|$ to $O(\text{OPT}(G)/\varepsilon)$, while w.h.p. preserving the

maximum matching size to within a factor of $(1-\varepsilon)$. In the sparsified graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, although we only have $\text{OPT}(G_p) = \Omega(|\mathcal{V}|)$ instead of having a perfect matching, we can show that, with some more care in the analysis, the constant gap between $\text{OPT}(G_p)$ and $|\mathcal{V}|$ is enough for us to establish the approximation ratios of our algorithms.

Comparison with [Blum et al. 2015]. The adaptive algorithm of [Blum et al. 2015] can be summarized as follows: maintain a matching M, and at each round find a collection of vertex-disjoint *augmenting paths* of length $O(1/\varepsilon)$ in the input graph $G(V,E)$ with respect to M; query the edges of the augmenting paths and augment M if possible. Using the well-known fact that a matching with no augmenting path of length $O(1/\varepsilon)$ is a $(1-\varepsilon)$-approximate matching, the authors show that the found matching of the algorithm is a $(1-\varepsilon)$-approximation in expectation.

In this process, the probability that an augmenting path of length $O(1/\varepsilon)$ "survives" the querying process is only $p^{O(1/\varepsilon)}$; hence, one needs to repeat the whole process roughly $\frac{1}{p^{O(1/\varepsilon)}}$ times, which leads to the same degree of adaptivity and per-vertex queries. The non-adaptive algorithm of [Blum et al. 2015] is designed based on similar framework of using augmenting paths.

On the other hand, our algorithms exploit the structure of matchings in a global way (using the Tutte-Berge formula) instead of locally searching for short augmenting paths. In particular, through the use of matching covers, we completely eliminate the need of searching for the augmenting paths and hence avoid the exponential dependency on $\frac{1}{\varepsilon}$, which is essentially the length of the augmenting paths. It is worth mentioning that however most of these differences only appear in the analysis; the description of our algorithms and algorithms of [Blum et al. 2015] are both simple and similar (modulo the extra sparsification part of our algorithm).

2. PRELIMINARIES

Notation. Throughout we use n to denote $|V|$, where V is the set of vertices in the input graph $G(V,E)$. For any set of edges $X \subseteq E$, $V(X)$ denotes the set of vertices incident on X. For two integers $a \leq b$, $[a,b]$ denotes the set $\{a,\ldots,b\}$ and $[b] := [1,b]$.

Tutte-Berge formula. In our proofs, we crucially rely on the Tutte-Berge formula which generalizes the *Hall's marriage theorem* for characterizing perfect matchings in bipartite graphs to maximum matchings in general graphs. For any graph $G(V,E)$ and any $U \subseteq V$, $odd(V-U)$ denotes the number of connected components with *odd* number of vertices in $G(V \setminus U, E)$. We have,

LEMMA 2.1 (TUTTE-BERGE FORMULA). *The size of a maximum matching in a graph $G(V,E)$ is equal to*

$$\frac{1}{2}\min_{U\subseteq V}\Big(|U|+|V|-odd(V-U)\Big)$$

See, e.g., [Lovász and Plummer 2009] (Chapter 3) for a proof of this lemma.

Finally, we have the following simple concentration result on the size of a maximum matching in G_p; the proof is a standard application of the Chernoff bound.

CLAIM 2.2. *For any graph $G(V,E)$ with $\text{OPT}(G) = \omega(1/p)$,*

$$\Pr\left(\text{OPT}(G_p) \geq p\cdot\text{OPT}(G)/2\right) = 1-o(1)$$

3. MATCHING COVERS AND VERTEX SPARSIFICATION FOR STOCHASTIC MATCHING

In this section, we present our main technical results, namely the *matching-cover lemma* and the *vertex sparsification lemma*, which lie in the core of both our adaptive and non-adaptive algorithms. We start by describing the matching-cover lemma.

As explained earlier in Section 1.5, the matching-cover lemma is already sufficient for establishing the approximation guarantee of the algorithms as long as $\text{OPT}(G) = \Omega(n)$. To tackle the case where $\text{OPT}(G)$ is much smaller than the number of vertices, we next introduce a simple vertex sparsification lemma that provides a way of reducing the number of vertices in any graph $G(V, E)$ from $|V|$ to $O(\text{OPT}(G))$ while preserving the maximum matching size approximately, for *any* realization of $G(V, E_p)$.

3.1. Matching-Cover Lemma

We start by defining the following process which takes any graph as an input, and outputs a list of matchings (i.e., a *matching-cover*).

Definition 3.1 (*Incremental Matching Selection Process*). We say an algorithm is an *incremental matching selection process (IMSP)* iff for any input graph G, the algorithm selects a sequence of matchings $M_1, M_2, \ldots, M_r$ one by one from G such that for any $i \in [r]$, for any edge e selected in the i-th matching M_i where e also appears in M_j for some $j < i$, the edge e must be realized.

We refer to the matchings an IMSP outputs as a *matching-cover*, and we say that an IMSP outputs a *γ-matching-cover* iff for all $i \in [r]$, $|M_i| \geq \frac{\gamma n}{2}$. The following claim states the key property of any IMSP, which will be used in our proofs.

CLAIM 3.2. *For any IMSP A and any graph G, denote by $M_1, M_2, \ldots, M_r$ the matching-cover that A outputs on G; then for any $i \in [r]$, conditioned on any realization of $M_1, M_2, \ldots, M_{i-1}$, and any choice of the matching M_i, each edge $e \in M_i$ is realized w.p. at least p, independent of any other edges in M_i.*

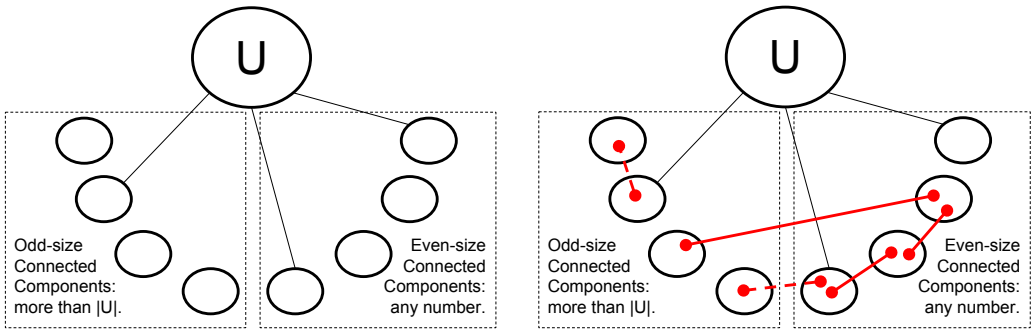
PROOF. For the edges $e \in M_i$ that appear in previous matchings, by the definition of IMSP, e is realized w.p. $1 (\geq p)$, which is trivially independent of any other edges in M_i. For the set of edges E' that do not appear in any previous matching, E' is disjoint with $M_1, M_2, \ldots, M_{i-1}$ and the set of realized edges in $M_1, M_2, \ldots, M_{i-1}$ is independent of realization of edges in E'. Therefore, by the definition of the stochastic setting, each edge in E' is realized w.p. p, independent of other edges. □

We are now ready to state the matching-cover lemma, which is the main result of this section.

LEMMA 3.3 (MATCHING-COVER LEMMA). *For any parameter $0 < \varepsilon, p < 1$, any graph G, and any IMSP A, denote by $M_1, \ldots, M_r$ the γ-matching-cover of G that A outputs. If $r \geq \frac{32 \log(2e/\gamma)}{\varepsilon p}$ and $\gamma \cdot n = \omega(1)$, then, w.p. $1 - o(1)$, there is a matching of size at least $(1 - \varepsilon)\frac{\gamma n}{2}$ among the realized edges in the matching-cover.*

We remark that Lemma 3.3 holds even for multi-graphs, which is a property required by our algorithms (due to the usage of vertex sparsification). We first provide a high-level summary of the proof. Suppose by contradiction that the output of IMSP A does not contain a large matching; then, by Tutte-Berge formula, there should exist a set of vertices $U \subseteq V$ where removing U from the graph results in many connected components (CC) with odd number of vertices, namely U is an *odd-sets-witness* (see Fig 1.a for an example of an odd-sets-witness). Our strategy is to show that, for any fixed set U, the probability that U ends up being an odd-sets-witness is sufficiently small, and then use a union bound over all possible choices of U to argue that w.h.p. no such odd-sets-witness can arise.

To see that w.h.p, each U does not lead to an odd-sets-witness, we again use the Tutte-Berge formula: if the edges realized in the first i matchings leave many odd-size CC's, then the large matching M_{i+1} must eliminate most of them. Note that this is not enough to show that the number of *odd-size CC's* will decrease w.h.p. since it is



(a) A candidate odd-sets-witness U.

(b) Thick (red) edges are matching M_i. Solid edges are realized while dashed edges are not.

Fig. 1: An example of adding a large matching M_i to an odd-sets-witness (the red/thick edges). The number of odd components does not decrease but the total number of connected components indeed decreases.

possible that M_{i+1} eliminates two odd-size CC's by connecting them through a chain of even-size CC's (see the long chain in Fig 1.b for an illustration). The length of the chain could be arbitrarily long and even if one edge on the chain is not realized, the two odd-size CC's will still end up being disconnected. A key observation here is that though the number of odd-size CC's might not decrease (requiring all edges on the chain to be realized), but the total number of CC's will decrease w.h.p. (any realized edge on the chain reduces the number of CC's). Using this fact, we show that after enough number of rounds, the total number of CC's will drop significantly and even if all of them are odd-size CC's, it is not enough for being a odd-sets-witness.

PROOF OF LEMMA 3.3. If the edges realized in the matching-cover do not contain a matching of size more than $(1-\varepsilon)\frac{\gamma n}{2}$, then, by Lemma 2.1, there exists a set of vertices U where the number of odd-size connected components after removing U, i.e., $odd(V-U)$, satisfies

$$(1-\varepsilon)\frac{\gamma n}{2} \geq \frac{1}{2}\Big(|U|+|V|-odd(V-U)\Big)$$

which implies that $odd(V-U) \geq |U| + (1-\gamma+\varepsilon\gamma)n$. In this case, we say U leads to an *odd-sets-witness* and denote this event by E_U. Using this fact, we only need to prove the following lemma.

LEMMA 3.4. *For any $U \subseteq V$,* $\Pr(\mathsf{E}_U) \leq 2^{-2\gamma n \log(2e/\gamma)}$.

We first show that Lemma 3.4 implies Lemma 3.3. We will apply a union bound over all candidate sets U to show that probability that *there exists* some U where E_U happens is $o(1)$. In order to so, we argue that the number of different choices of U's that need to be considered is at most $2^{\gamma n \log(2e/\gamma)}$. To see this, note that every odd-size set must contain at least one vertex, and there are only n vertices that could be part of the odd-size sets. Thus, we have $n \geq odd(V-U)$. On the other hand, as we just established, $odd(V-U) \geq |U| + (1-\gamma+\varepsilon\gamma)n$, and combining the two inequalities, we have

$$|U| \leq n - (1-\gamma+\varepsilon\gamma)n \leq \gamma n \tag{1}$$

Therefore, it suffices to consider the sets U with cardinality at most γn, and only

$$\binom{n+\gamma n}{\gamma n} \leq \left(\frac{e(1+\gamma)n}{\gamma n}\right)^{\gamma n} \leq \left(\frac{2e}{\gamma}\right)^{\gamma n} = 2^{\gamma n \log(2e/\gamma)}$$

such choices of U exists. Now, we can apply a union bound over all such choices of U, and by Lemma 3.4 w.p. at least $1 - 2^{-\gamma n \log(2e/\gamma)} = 1 - o(1)$ (recall that $\gamma n = \omega(1)$), there is no odd-sets-witness, proving Lemma 3.3. It remains to prove Lemma 3.4, and as stated in Eq (1) we can assume, wlog, that $|U| \leq \gamma n$.

PROOF OF LEMMA 3.4. Recall that the goal is to show that w.h.p., U does not lead to an odd-sets-witness (i.e., E_U does not happen). We first define some notation. For any $i \in [r]$, $M_p(i)$ denotes the set of edges in M_i that are realized and are not incident on vertices in U, and G_i denotes the graph after realizing the edges in the first i matchings. We use $cc(G_i)$ to denote the number of connected components in G_i and use E_i to denote the event that the number of odd-size connected components in G_i is at least $|U| + (1 - \gamma + \varepsilon\gamma)n$.

Let Y_i be a random binary variable where $\mathsf{Y}_i = 1$ iff $cc(G_{i-1}) - cc(G_i) < \varepsilon p \cdot \gamma n/16$ (and $\mathsf{Y}_i = 0$ otherwise), which is the event that the edges of $M_p(i)$ do not reduce the number of connected components in G_{i-1} by more than $\varepsilon p \cdot \gamma n/16$. Suppose for at least half of the rounds, $\mathsf{Y}_i = 0$; then, the number of connected components after the IMSP selects the r matchings is less than

$$n - \frac{r}{2} \cdot \frac{\varepsilon p \cdot \gamma n}{16} \leq n - \frac{32\log(2e/\gamma)}{\varepsilon p} \cdot \frac{1}{2} \cdot \frac{\varepsilon p \cdot \gamma n}{16} \leq (1-\gamma)n.$$

On the other hand, since having $|U| + (1-\gamma+\varepsilon\gamma)n$ odd-size connected components (i.e., E_U happens) implies that there are more than $(1-\gamma)n$ connected components,

$$\mathbf{Pr}\left(\mathsf{E}_U\right) = \mathbf{Pr}\left(\mathsf{E}_U, \sum_{i\in[r]} \mathsf{Y}_i \geq \frac{r}{2}\right) \tag{2}$$

Hence, we can focus on upper bounding the probability that E_U happens *and* for more than half of the rounds, $\mathsf{Y}_i = 1$. In the following, we first establish a key property regarding the probability that $\mathsf{Y}_i = 1$ (Lemma 3.5) and then show how to use this property to bound the probability of the target event in Eq (2) (Lemma 3.7). To simplify the presentation, we use $M_p(i_i, i_2, \ldots, i_j)$ to denote the set of edges realized in the matchings $M_p(i_1), M_p(i_2), \ldots, M_p(i_j)$, and, with a slight abuse of notation, use Y_i to denote the event that $\mathsf{Y}_i = 1$. In particular, we show that

LEMMA 3.5. *For any $M_p(1, \ldots, i-1)$,*

$$\mathbf{Pr}\left(\mathsf{Y}_i, \mathsf{E}_{i-1} \mid M_p(1,\ldots,i-1)\right) \leq \exp\left(-\frac{3\varepsilon p \cdot \gamma n}{16}\right)$$

PROOF. Recall that E_{i-1} is the event that the number of odd-size connected components in G_{i-1} is at least $|U| + (1-\gamma+\varepsilon\gamma)n$. We have,

$$\begin{aligned}
&\mathbf{Pr}\left(\mathsf{Y}_i, \mathsf{E}_{i-1} \mid M_p(1,\ldots,i-1)\right) \\
=&\mathbf{Pr}\left(\mathsf{Y}_i \mid M_p(1,\ldots,i-1), \mathsf{E}_{i-1}\right) \cdot \mathbf{Pr}\left(\mathsf{E}_{i-1} \mid M_p(1,\ldots,i-1)\right) && \text{(Chain rule)} \\
\leq&\mathbf{Pr}\left(\mathsf{Y}_i \mid M_p(1,\ldots,i-1), \mathsf{E}_{i-1}\right)
\end{aligned}$$

hence, we only need to show that $\mathbf{Pr}\left(\mathsf{Y}_i \mid M_p(1,\ldots,i-1), \mathsf{E}_{i-1}\right) \leq \exp\left(-\frac{3\varepsilon p\cdot\gamma n}{16}\right)$, which is, roughly speaking, the probability that the i-th matching M_i does not reduce the number of connected components by a lot, given that the graph G_{i-1} contains many odd-size connected components.

To proceed, we need the following definition. For any graph H, we say a set of edges E' form a *component-based spanning forest* of H, if E' is a spanning forest of the graph obtained by contracting each connected component in H into a single vertex (and any edge $(u,v) \in E'$ becomes an edge between the connected components that u and v respectively resides in). It is straightforward to verify that if we add any component-based spanning forest $E' \subset E$ to H, the number of connected components in H would reduce by $|E'|$. The following claim is the key to obtain the target upper bound on $\mathbf{Pr}\left(\mathsf{Y}_i \mid M_p(1,\ldots,i-1), \mathsf{E}_{i-1}\right)$.

CLAIM 3.6. *Whenever E_{i-1} happens, there exist at least $\frac{\varepsilon\gamma n}{2}$ edges of M_i that form a component-based spanning forest of G_{i-1}.*

PROOF. Since the matching M_i has size at least $\gamma n/2$ (by definition of being in a γ-matching-cover), the edges of M_i can reduce the number of odd-size sets from at least $|U| + (1-\gamma+\varepsilon\gamma)n$ (i.e., E_{i-1} happens) down to at most $|U| + (1-\gamma)n$ (by Lemma 2.1). Therefore, after adding edges of M_i to G_{i-1}, at least $\varepsilon\gamma n$ odd-size sets will disappear. For each odd-size set S that disappears, M_i must contain at least one edge between S and another connected component. Therefore, for the largest component-based spanning forest obtained by the edges in M_i, at least $\varepsilon\gamma n$ vertices (i.e., connected components) have degree at least one, which implies the number of edges in the forest is at least $\frac{\varepsilon\gamma n}{2}$. □

Let $T_i \subseteq M_i$ be the the set of (at least) $\frac{\varepsilon\gamma n}{2}$ edges promised by Claim 3.6 (conditioned on E_{i-1}) and t_i be the number of edges realized in T_i. By Claim 3.2, each edge in T_i is realized w.p. at least p independent of each other, hence, $\mathbf{E}\left[t_i \,|\, \mathsf{E}_{i-1}\right] \geq \varepsilon p \cdot \gamma n/2$. Note that t_i is a lower bound on $cc(G_{i-1}) - cc(G_i)$ (i.e., the decrement of the number of connected components from G_{i-1} to G_i), and $\mathsf{Y}_i = 1$ iff $cc(G_{i-1}) - cc(G_i) \leq \varepsilon p \cdot \gamma n/16$, which implies that no more than $\varepsilon p \cdot \gamma n/16$ edges is realized in T_i (which is $1/8$ of the expectation). Hence, by Chernoff bound,

$$\begin{aligned}
&\mathbf{Pr}\left(\mathsf{Y}_i \mid M_p(1,\ldots,i-1), \mathsf{E}_{i-1}\right) \leq \mathbf{Pr}\left(t_i \leq \frac{\varepsilon p \cdot \gamma n}{16} \mid \mathsf{E}_{i-1}\right) \\
\leq &\exp\left(-\frac{1}{2}\cdot\left(\frac{7}{8}\right)^2 \cdot \frac{\varepsilon p\cdot\gamma n}{2}\right) = \exp\left(-\frac{49}{64}\cdot\frac{\varepsilon p\cdot\gamma n}{4}\right) \\
\leq &\exp\left(-\frac{48}{64}\cdot\frac{\varepsilon p\cdot\gamma n}{4}\right) \leq \exp\left(-\frac{3\varepsilon p\cdot\gamma n}{16}\right)
\end{aligned}$$

which concludes the proof of Lemma 3.5. □

Having Lemma 3.5, we are now ready to upper bound the probability that Y_i happens in more than half of the rounds.

LEMMA 3.7. *For any collection of $r/2$ rounds $i_1, i_2, \ldots, i_{r/2}$,*

$$\mathbf{Pr}\left(\mathsf{Y}_{i_1}, \mathsf{Y}_{i_2}, \ldots, \mathsf{Y}_{i_{r/2}}, \mathsf{E}_U\right) \leq 2^{-3\gamma n \log(2e/\gamma)}.$$

PROOF. Assume wlog that $i_1 < i_2 < \ldots < i_{r/2}$. First of all, E_U happening implies that $\mathsf{E}_{i_1-1}, \mathsf{E}_{i_2-1}, \ldots, \mathsf{E}_{i_{r/2}-1}$ should all happen[3]; therefore,

$$\mathbf{Pr}\left(\mathsf{Y}_{i_1}, \mathsf{Y}_{i_2}, \ldots, \mathsf{Y}_{i_{r/2}}, \mathsf{E}_U\right) \leq \mathbf{Pr}\left(\mathsf{Y}_{i_1}, \mathsf{Y}_{i_2}, \ldots, \mathsf{Y}_{i_{r/2}}, \mathsf{E}_{i_1-1}, \mathsf{E}_{i_2-1}, \ldots, \mathsf{E}_{i_{r/2}-1}\right) \qquad (3)$$

[3]It is straightforward to very that the number of odd-size connected components is monotonically decreasing.

After reorganizing the terms, we have,

$$\begin{aligned}
&\mathbf{Pr}\left(\mathsf{Y}_{i_1}, \mathsf{Y}_{i_2}, \ldots, \mathsf{Y}_{i_{r/2}}, \mathsf{E}_{i_1-1}, \mathsf{E}_{i_2-1}, \ldots, \mathsf{E}_{i_{r/2}-1}\right)\\
=&\mathbf{Pr}\left(\mathsf{Y}_{i_1}, \mathsf{E}_{i_1-1}, \mathsf{Y}_{i_2}, \mathsf{E}_{i_2-1}, \ldots, \mathsf{Y}_{i_{r/2}}, \mathsf{E}_{i_{r/2}-1}\right)\\
=&\Pi_{j\in[r/2]}\,\mathbf{Pr}\left(\mathsf{Y}_{i_j}, \mathsf{E}_{i_j-1} \mid \mathsf{Y}_{i_1}, \mathsf{E}_{i_1-1}, \mathsf{Y}_{i_2}, \mathsf{E}_{i_2-1}, \ldots, \mathsf{Y}_{i_{j-1}}, \mathsf{E}_{i_{j-1}-1}\right) \qquad \text{(Chain rule)}
\end{aligned}$$

We will upper bound each of the $r/2$ terms separately. Fix a $j \in [r/2]$, denote the event $(\mathsf{Y}_{i_1}, \mathsf{E}_{i_1-1}, \mathsf{Y}_{i_2}, \mathsf{E}_{i_2-1}, \ldots, \mathsf{Y}_{i_{j-1}}, \mathsf{E}_{i_{j-1}-1})$ by E^*. Note that E^* is completely determined by $M_p(1, \ldots, i_{j-1})$, which is also determined by $M_p(1, \ldots, i_j - 1)$ since $i_j - 1 \geq i_{j-1}$. We have

$$\begin{aligned}
&\mathbf{Pr}\left(\mathsf{Y}_{i_j}, \mathsf{E}_{i_j-1} \mid \mathsf{Y}_{i_1}, \mathsf{E}_{i_1-1}, \mathsf{Y}_{i_2}, \mathsf{E}_{i_2-1}, \ldots, \mathsf{Y}_{i_{j-1}}, \mathsf{E}_{i_{j-1}-1}\right) = \mathbf{Pr}\left(\mathsf{Y}_{i_j}, \mathsf{E}_{i_j-1} \mid \mathsf{E}^*\right)\\
=&\sum_{M_p(1,\ldots,i_j-1)} \mathbf{Pr}\left(M_p(1, \ldots, i_j - 1) \mid \mathsf{E}^*\right) \cdot \mathbf{Pr}\left(\mathsf{Y}_{i_j}, \mathsf{E}_{i_j-1} \mid \mathsf{E}^*, M_p(1, \ldots, i_j - 1)\right)\\
=&\sum_{M_p(1,\ldots,i_j-1) \text{ s.t. } \mathsf{E}^* happens} \mathbf{Pr}\left(M_p(1, \ldots, i_j - 1) \mid \mathsf{E}^*\right) \cdot \mathbf{Pr}\left(\mathsf{Y}_{i_j}, \mathsf{E}_{i_j-1} \mid M_p(1, \ldots, i_j - 1)\right)\\
&\qquad\qquad (\mathsf{E}^* \text{ is determined by } M_p(1, \ldots, i_j - 1))\\
\leq&\sum_{M_p(1,\ldots,i_j-1) \text{ s.t. } \mathsf{E}^* happens} \mathbf{Pr}\left(M_p(1, \ldots, i_j - 1) \mid \mathsf{E}^*\right) \cdot \exp\left(-\frac{3\varepsilon p \cdot \gamma n}{16}\right) \quad \text{(By Lemma 3.5)}\\
=&\exp\left(-\frac{3\varepsilon p \cdot \gamma n}{16}\right)
\end{aligned}$$

Therefore

$$\mathbf{Pr}\left(\mathsf{Y}_{i_1}, \mathsf{Y}_{i_2}, \ldots, \mathsf{Y}_{i_{r/2}}, \mathsf{E}_U\right) \leq \exp\Big(-\frac{3\varepsilon p \cdot \gamma n}{16} \cdot \frac{r}{2}\Big) \leq \exp(-3\gamma n \log(2e/\gamma)) \leq 2^{-3\gamma n \log(2e/\gamma)}$$

where the second equality is by the choice of r. □

By Lemma 3.7, for each collection of $\frac{r}{2}$ rounds, Y_i happens to all of them w.p. at most $2^{-3\gamma n \log(2e/\gamma)}$. There are at most 2^r (which is independent of n) choices of different (at least) $\frac{r}{2}$ rounds, hence using union bounds, for n sufficiently large, the prob. that Y_i happens in more than $\frac{r}{2}$ rounds is at most $2^{-2\gamma n \log(2e/\gamma)}$, proving Lemma 3.4. As discussed earlier, Lemma 3.4 implies Lemma 3.3, which completes the proof. □

3.2. Vertex Sparsification Lemma

In the following, we give an algorithm that for any $0 < \varepsilon < 1$, reduces the number of vertices in any graph G from $|V|$ to $O(\text{OPT}(G)/\varepsilon)$, while preserving the maximum matching size to within a factor of $(1 - \varepsilon)$ for *any* realization G_p of G w.h.p.

For inputs G and ε and a sparsification parameter τ to be determined later, SPARSIFY(G, τ, ε) works as follows. We create and output a multi-graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where: (i) $|\mathcal{V}| = \tau$, (ii) each vertex v in G is mapped to a vertex $\mathcal{V}(v)$ in $\mathcal{G}$ chosen *uniformly at random* from $\mathcal{V}$, and (iii) for each edge (u, v), there is a corresponding edge between $\mathcal{V}(u)$ and $\mathcal{V}(v)$. A pseudo-code of the SPARSIFY algorithm is presented in Algorithm 1. We point out that similar ideas of randomly grouping vertices for matchings have been also recently used in [Assadi et al. 2016; Chitnis et al. 2016] for the purpose of reducing *space requirement* of algorithms in graph streams.

The following lemma states the main property of the SPARSIFY algorithm.

LEMMA 3.8. *For any graph $G(V, E)$, suppose $\mathcal{G}(\mathcal{V}, \mathcal{E}) :=$ SPARSIFY(G, τ, ε) for the parameter $\tau \geq \frac{4 \cdot \text{OPT}(G)}{\varepsilon}$, and let M be any fixed matching in G with $|M| = \omega(1)$; then,*

ALGORITHM 1: SPARSIFY(G, τ, ε). A Matching-Preserving Sparsification Algorithm

Input: Graph $G(V, E)$, sparsification parameter τ, and input parameter $\varepsilon > 0$.
Output: A multi-graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = \tau$.

(1) Partition the vertices in V into τ groups $\mathcal{V} := (\mathcal{V}_1, \ldots, \mathcal{V}_\tau)$, by assigning each vertex *independently* to one of the τ groups chosen *uniformly at random*.
(2) For any edge $(u, v) \in E$, add an edge $e_{u,v}$ between the vertices $\mathcal{V}(u)$ and $\mathcal{V}(v)$ in $\mathcal{E}$.
(3) Return the multi-graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

w.p. $1 - o(1)$*, there exists a matching* $\mathcal{M}$ *of size* $(1 - \varepsilon) \cdot |M|$ *in* $\mathcal{G}$*. Moreover, the edges of* $\mathcal{M}$ *correspond to a unique matching* M' *in* G *of the same size.*

PROOF. Let $\varepsilon' := \varepsilon/4$, and let s denote the number of vertices matched in M (i.e., $s = 2|M|$). Note that $\tau \geq 4\text{OPT}(G)/\varepsilon \geq 4s/\varepsilon = s/\varepsilon'$. For the sake of analysis, we first merge the τ groups $\mathcal{V}$ into s/ε' *super-groups* in the following manner. Fix any partition that evenly breaks $[\tau]$ into s/ε' non-empty parts $P_1, P_2, \ldots, P_{s/\varepsilon'}$ (i.e., $|P_i| = \frac{\varepsilon'\tau}{s}$ for any $i \in [s/\varepsilon']$). For each partition P_i, define a super-group $\mathcal{S}_i := \cup_{j \in [P_i]} \mathcal{V}_j$. Then, since in Algorithm 1 each vertex $v \in V$ is assigned to exactly one group chosen uniformly at random, the probability that $v \in \mathcal{S}_i$ is $\frac{\varepsilon'}{s}$.

We say a super-group $\mathcal{S}_i$ is *good* iff $\mathcal{S}_i$ contains at least one vertex in $V(M)$, and is otherwise *bad*. For each super-group $\mathcal{S}_i$, let X_i be a random variable where $X_i = 1$ iff $\mathcal{S}_i$ is bad (otherwise, $X_i = 0$). Let $X := \sum_{i \in [s/\varepsilon']} X_i$ be the number of bad super-groups. In the following, we first show that X is small w.h.p., which implies that there are many good super-groups, and then show that there is a large matching between the good super-groups.

To see that $X = \sum_{i \in [s/\varepsilon']} X_i$ is small w.h.p., first notice that

$$\mathbf{Pr}(X_i = 1) = \left(1 - \frac{\varepsilon'}{s}\right)^s \leq e^{-\varepsilon'} \leq 1 - \varepsilon' + \frac{\varepsilon'^2}{2} \qquad (\forall x \geq 0, e^{-x} \leq 1 - x + x^2/2)$$

Therefore, we have $\mathbf{E}[X] = \sum_{i \in [s/\varepsilon']} \mathbf{E}[X_i] \leq (1 - \varepsilon' + \frac{\varepsilon'^2}{2})\frac{s}{\varepsilon'}$. On the other hand, since at most $|V(M)|\,(= s)$ super-groups could contain a vertex from $V(M)$ (i.e., could be good), $X \geq s/\varepsilon' - s = \Omega(s) = \omega(1)$, and hence $\mathbf{E}[X] = \omega(1)$. To continue, observe that our setting can be viewed as a standard balls and bins experiment: each vertex in $V(M)$ is a ball; each super-group is a bin; and X_i denotes the event that the i-th bin is empty. Therefore, the random variables X_i's are *negatively correlated*, and we can apply Chernoff bound [Panconesi and Srinivasan 1997]:

$$\mathbf{Pr}\left(X \geq (1 + \varepsilon'^2)\,\mathbf{E}[X]\right) \leq e^{-\Omega_{\varepsilon'}(\mathbf{E}[X])} = o(1) \qquad (\mathbf{E}[X] = \omega(1))$$

Since $\mathbf{E}[X] \leq (1 - \varepsilon' + \frac{\varepsilon'^2}{2})\frac{s}{\varepsilon'}$ (as shown above), we further have

$$\mathbf{Pr}\left(X \geq (1 + \varepsilon'^2)(1 - \varepsilon' + \frac{\varepsilon'^2}{2})\frac{s}{\varepsilon'}\right) \leq \mathbf{Pr}\left(X \geq (1 + \varepsilon'^2)\,\mathbf{E}[X]\right) = o(1)$$

Hence, w.p. $1 - o(1)$, the number of bad super-groups is at most $(1 - \varepsilon' + 2\varepsilon'^2)\frac{s}{\varepsilon'}$, which implies that the number of good super-groups is at least $\frac{s}{\varepsilon'} - (1 - \varepsilon' + 2\varepsilon'^2)\frac{s}{\varepsilon'} = (\varepsilon' - 2\varepsilon'^2)\frac{s}{\varepsilon'} = (1 - 2\varepsilon')s$.

It remains to show that if at least $(1 - 2\varepsilon')s$ super-groups are good (i.e., contain a vertex from $V(M)$), then the edges in M form a matching $\mathcal{M}$ in $\mathcal{G}$ of size at least $(1 - 4\varepsilon')\,|M|\,(= (1 - \varepsilon)\,|M|)$. To see this, for each good super-group $\mathcal{S}_i$, we fix one vertex $v \in V(M) \cap \mathcal{S}_i$ and remove all other vertices in $\mathcal{S}_i$. For the matching M, at most $2\varepsilon' s$

vertices in $V(M)$ are removed and hence at least $s/2 - 2\varepsilon' s = (1-4\varepsilon')\,|M|$ edges in M remain. Since all endpoints of these edges are assigned to distinct super-groups, these edges form a matching $\mathcal{M}$ of size at least $(1-4\varepsilon')\,|M| = (1-\varepsilon)\,|M|$ in $\mathcal{G}$.

To see the second part of the lemma, simply note that each edge of $\mathcal{M}$ comes from a distinct edge in M. □

For any $G_p := G(V, E_p)$, define $\mathcal{G}_p$ as the graph obtained from $\mathcal{G}$ by considering only the edges that correspond to edges in E_p. We are now ready to prove our vertex sparsification lemma.

LEMMA 3.9 (VERTEX SPARSIFICATION LEMMA). *Let $G(V,E)$ be a graph with maximum matching size $\omega(1/p)$ and let $\mathcal{G} = \mathsf{SPARSIFY}(G, \tau, \varepsilon)$ for $\tau = \frac{4\cdot \mathrm{OPT}(G)}{\varepsilon}$; then,*

$$\Pr\Big(\textsc{opt}(\mathcal{G}_p) \geq (1-\varepsilon)\cdot \textsc{opt}(G_p)\Big) = 1 - o(1)$$

where the probability is taken over both the inner randomness of SPARSIFY *algorithm as well as the realization $E_p \subseteq E$.*

PROOF. By Claim 2.2, the maximum matching size in G_p is $\omega(1)$ w.p. $1-o(1)$. Now, for any realization G_p with maximum matching size of $\omega(1)$, by construction, $\mathcal{G}_p = \mathsf{SPARSIFY}(G_p, \tau, \varepsilon)$. Hence, by applying Lemma 3.8 on any maximum matching M in G_p, we have that $\mathcal{G}_p$ has a matching of size $(1-\varepsilon)\,|M|$ w.p. $1-o(1)$. □

4. A $(1-\varepsilon)$-APPROXIMATION ADAPTIVE ALGORITHM

We now present our adaptive algorithm and prove Theorem 1.1. The following is a formal restatement of Theorem 1.1.

THEOREM 4.1. *There is an* adaptive *algorithm that for any input graph $G(V,E)$, and any input parameter $\varepsilon > 0$, outputs a matching of size $\textsc{alg} := \textsc{alg}(G_p)$ such that,*

$$\Pr\Big(\textsc{alg} \geq (1-\varepsilon)\cdot \textsc{opt}\Big) = 1 - o(1)$$

where $\textsc{opt} := \textsc{opt}(G_p)$ is the maximum matching size in $G_p(V, E_p)$. The probability is taken over both the inner randomness of the algorithm and the realization $E_p \subseteq E$.

Moreover, the algorithm makes only $O(\frac{\log(1/\varepsilon p)}{\varepsilon p})$ rounds of adaptive queries, and queries only $O(\frac{\log(1/\varepsilon p)}{\varepsilon p})$ edges per vertex.

Our adaptive algorithm in Theorem 4.1 works as follows. We first use our vertex sparsification lemma (Lemma 3.9) to compute a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}) := \mathsf{SPARSIFY}(G, \tau, \varepsilon)$ where $\textsc{opt}(\mathcal{G}) = \Omega(|\mathcal{V}|)$. Next, we repeat for $O(\frac{\log(1/\varepsilon p)}{\varepsilon p})$ rounds the following operation. Pick a maximum matching M from $\mathcal{G}$, query the edges of M, and remove the edges that are not realized. Finally, return a maximum matching among the realized edges. The pseudo-code of this algorithm is presented as Algorithm 2.

It is straightforward to verify the number of queries and the degree of adaptivity used in Algorithm 2: each round of the algorithm queries edges of a matching and hence each vertex is queried at most once in each round. We now prove the bound on the approximation ratio of the algorithm.

PROOF OF THEOREM 4.1. Let $\widehat{\textsc{opt}} := \widehat{\textsc{opt}}(\mathcal{G}_p)$ denote the maximum matching size in the graph $\mathcal{G}_p$. By Lemma 3.9, w.p. $1-o(1)$, $\widehat{\textsc{opt}} \geq (1-\varepsilon)\textsc{opt}(G_p)$. Now, it suffices to show that w.p. $1-o(1)$, Algorithm 2 outputs a matching of size at least $(1-\varepsilon)\widehat{\textsc{opt}}$,

ALGORITHM 2: A $(1-\varepsilon)$-Approximation Adaptive Algorithm for Stochastic Matching

Input: Graph $G(V,E)$ and input parameters $0 < \varepsilon, p < 1$.
Output: A matching M in $G(V,E_p)$.

(1) Let $\mathcal{G}(\mathcal{V},\mathcal{E}) := \textsf{SPARSIFY}(G,\tau,\varepsilon)$ for $\tau := \left\lceil \frac{4\text{OPT}(G)}{\varepsilon} \right\rceil$.
(2) Let $R := \left\lceil \frac{32\log(8e/\varepsilon p)}{\varepsilon \cdot p} \right\rceil$, and $\mathcal{E}^* \leftarrow \mathcal{E}$.
(3) For $i = 1, \ldots, R$, do:
  (a) Pick a *maximum matching* M_i in $\mathcal{G}(\mathcal{V},\mathcal{E}^*)$.
  (b) *Query* the edges in M_i and remove the *non-realized edges* from $\mathcal{E}^*$.
(4) Output a maximum matching among *realized* edges in $M_1, M_2, \ldots, M_R$.

since, with a union bound, it would imply w.p. $1-o(1)$,

$$\text{ALG}(G_p) \geq (1-\varepsilon)\cdot\widehat{\text{OPT}} \geq (1-\varepsilon)^2\cdot\text{OPT}(G_p) \geq (1-2\varepsilon)\cdot\text{OPT}(G_p)$$

and we can replace ε with $\varepsilon/2$ in Algorithm 2 to obtain a $(1-\varepsilon)$-approximation.

To see that $\text{ALG}(G_p) \geq (1-\varepsilon)\widehat{\text{OPT}}$ w.h.p., let $L := \min_{i\in[R]}|M_i|$, i.e., the minimum size of a matching chosen by Algorithm 2. Since all M_i's are maximum matchings in $\mathcal{E}^*$ while $\mathcal{E}^*$ always contains all edges of the optimum matching in $\mathcal{G}_p$, we have $L \geq \widehat{\text{OPT}}$ and we can focus on showing $\text{ALG}(G_p) \geq (1-\varepsilon)L$.

It is straightforward to verify that the way Algorithm 2 selects the matchings $M_1, M_2, \ldots, M_R$ satisfies the condition of IMSP (Definition 3.1). Moreover, by Claim 2.2 and Lemma 3.9, we have, $\Pr\left(\widehat{\text{OPT}} \geq p\cdot\text{OPT}(G)/2\right) = 1-o(1)$. Therefore, we can use Lemma 3.3 with parameters:

$$\gamma = \frac{2L}{|\mathcal{V}|} \geq \frac{\varepsilon\cdot p}{4} \quad , \quad r = \frac{32\log(2e/\gamma)}{\varepsilon p} \leq \frac{32\log(8e/\varepsilon p)}{\varepsilon p} \leq R$$

which states that w.p. $1-o(1)$, the realized edges in matchings $M_1, M_2, \ldots, M_R$ contain a matching of size at least $(1-\varepsilon)\frac{\gamma|\mathcal{V}|}{2} = (1-\varepsilon)L$, which completes the proof. □

5. A $(\frac{1}{2}-\varepsilon)$-APPROXIMATION NON-ADAPTIVE ALGORITHM

In this section, we present our non-adaptive algorithm and prove Theorem 1.2. The following is a formal restatement of Theorem 1.2.

THEOREM 5.1. *There is a* non-adaptive *algorithm that for any input graph $G(V,E)$, any input parameter $\varepsilon > 0$, outputs a matching of size* $\text{ALG} := \text{ALG}(G_p)$ *such that,*

$$\Pr\left(\text{ALG} \geq (\frac{1}{2}-\varepsilon)\cdot\text{OPT}\right) = 1-o(1)$$

where $\text{OPT} := \text{OPT}(G_p)$ *is the maximum matching size in $G_p(V,E_p)$. The probability is taken over both the inner randomness of the algorithm and the realization $E_p \subseteq E$.*

Moreover, the algorithm non-adaptively queries $O(\frac{\log(1/\varepsilon p)}{\varepsilon p})$ edges per vertex.

Note that any non-adaptive algorithm works in the following framework:

(1) Compute a subgraph $H(V,Q)$ of the input graph $G(V,E)$ for some $Q \subseteq E$.
(2) Query all edges in Q and compute a maximum matching in $H(V,Q_p)$.

Therefore, the main task of any non-adaptive algorithm is to choose a "good" subgraph H. Our non-adaptive algorithm chooses a subgraph H as follows. We first use our vertex sparsification lemma to compute a graph $\mathcal{G}(\mathcal{V},\mathcal{E}) := \textsf{SPARSIFY}(G,\tau,\varepsilon)$ where

$\text{OPT}(\mathcal{G}) = \Omega(|\mathcal{V}|)$. Next, we repeat for $O(\frac{\log(1/\varepsilon p)}{\varepsilon p})$ times the process of picking a maximum matching from $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and removing the edges of the matching from $\mathcal{E}$. Let Q be the set of edges in these matchings; the algorithm returns $H(V, Q)$ as the subgraph H. A pseudo-code of this algorithm is presented as Algorithm 3.

ALGORITHM 3: A Non-Adaptive $\left(\frac{1}{2} - \varepsilon\right)$-Approximation Algorithm for Stochastic Matching

Input: Graph $G(V, E)$ and input parameters $0 < \varepsilon, p < 1$.
Output: A matching M in $G(V, E_p)$.

(1) Let $\mathcal{G}(\mathcal{V}, \mathcal{E}) := \textsf{SPARSIFY}(G, \tau, \varepsilon)$ for $\tau := \left\lceil \frac{4\text{OPT}(G)}{\varepsilon} \right\rceil$.

(2) Let $R := \left\lceil \frac{32 \log(16e/\varepsilon p)}{\varepsilon \cdot p} \right\rceil$.

(3) Initially $Q \leftarrow \emptyset$. For $i = 1, \ldots, R$, do:
  (a) Pick a *maximum matching* M_i in $\mathcal{G}(\mathcal{V}, \mathcal{E} \setminus Q)$.
  (b) Let $Q \leftarrow Q \cup M_i$.

(4) *Query* all edges in Q and return a maximum matching in $M_1, M_2, \ldots, M_R$.

We now briefly provide the intuition behind Algorithm 3. Similar to the adaptive case, using the vertex sparsification lemma (Lemma 3.9), our task reduces to approximating the maximum matching in $\mathcal{G}_p$ (as opposed to G_p). For the adaptive case, our algorithm guarantees that every selected matching is of size at least $\text{OPT}(\mathcal{G}_p)$, which allows us to use the matching-cover lemma directly to complete the argument. However, Algorithm 3 does not have such a strong guarantee since it is non-adaptive. To address this issue, we establish a weaker guarantee which allows us to obtain a $1/2$-approximation. The idea is as follows. On one hand, if the smallest matching selected by the algorithm is of size at least $\text{OPT}(\mathcal{G}_p)/2$, we can still invoke the matching-cover lemma (Lemma 3.3) and have that Algorithm 3 outputs a matching of size $(1-\varepsilon)\text{OPT}(\mathcal{G}_p)/2$. On the other hand, we show that if the smallest selected matching has size less than $\text{OPT}(\mathcal{G}_p)/2$, then for *any* maximum matching M^* in $\mathcal{G}_p$, we must have selected in Q at least half the edges of M^*, which immediately results in a matching of size $\text{OPT}(\mathcal{G}_p)/2$.

We now present the formal proof. In the following, let $L := \min_{i \in [R]} |M_i|$, i.e., the minimum size of a matching chosen by Algorithm 3. Note that $L = |M_R|$ since the size of matchings chosen by the algorithm is a *non-increasing* sequence by construction.

LEMMA 5.2. *If* $L \geq p \cdot \text{OPT}(G)/4$*, then,* $\Pr\left(\text{ALG} \geq (1-\varepsilon) \cdot L\right) = 1 - o(1)$.

PROOF. Since $M_1, \ldots, M_R$ are edge disjoint matchings, the process of choosing $M_1, \ldots, M_R$ is an IMSP (Definition 3.1). Hence, by Lemma 3.3 with parameters:

$$\gamma = \frac{2L}{|\mathcal{V}|} \geq \frac{\varepsilon \cdot p}{8} \quad , \quad r = \frac{32 \log(2e/\gamma)}{\varepsilon p} \leq \frac{32 \log(16e/\varepsilon p)}{\varepsilon p} = R$$

there exists a matching of size $(1-\varepsilon) \cdot \frac{\gamma|\mathcal{V}|}{2} = (1-\varepsilon) \cdot L$ in Q_p w.p. $1 - o(1)$. Noting that any matching of $\mathcal{G}_p$ in Q_p corresponds to a matching in G_p completes the proof. □

We define $\widehat{\text{OPT}} := \widehat{\text{OPT}}(\mathcal{G}_p)$ as the maximum matching size in $\mathcal{G}_p$.

LEMMA 5.3. $\Pr\left(\text{ALG} \geq \left(\frac{1}{2} - \varepsilon\right) \cdot \widehat{\text{OPT}} \mid \widehat{\text{OPT}} > 2L\right) = 1$.

PROOF. Let M be any arbitrary matching in $\mathcal{G}$. We have, $|M| - |Q \cap M| \leq L$ since otherwise, for the matching $M' := M \setminus Q$, we have $|M'| = |M| - |Q \cap M| > L > |M_R|$, contradicting the fact that M_R is a maximum matching in the remaining graph.

Now let M be any maximum matching in $\mathcal{G}_p$. Since $\widehat{\text{OPT}} > 2L$, we have $|M| > 2L$. Consequently, $|Q \cap M| \geq |M| - L \geq |M| - |M|/2 = |M|/2$. Hence, at least half of the edges in M are also present in Q, implying that in this case, $\text{ALG} \geq \widehat{\text{OPT}}/2$ w.p. 1. $\square$

We now prove the bound on the approximation ratio of Algorithm 3.

LEMMA 5.4. $\Pr\left(\text{ALG} \geq (\frac{1}{2} - 2\varepsilon) \cdot \text{OPT}\right) = 1 - o(1)$.

PROOF. Recall that $\widehat{\text{OPT}} := \widehat{\text{OPT}}(\mathcal{G}_p)$. By Claim 2.2 and Lemma 3.9, we have,

$$\Pr\left(\widehat{\text{OPT}} \geq p \cdot \text{OPT}(G)/2\right) = 1 - o(1) \tag{4}$$

Let E_{win} be the event that $\text{ALG} \geq \left(\frac{1}{2} - \varepsilon\right) \cdot \widehat{\text{OPT}}$. We argue that $\Pr(\mathsf{E}_{win}) = 1 - o(1)$. This, together with the fact that w.p. $1-o(1)$, $\widehat{\text{OPT}} \geq (1-\varepsilon) \cdot \text{OPT}$ (Lemma 3.9) completes the proof.

Consider two cases, (i) $L < p \cdot \text{OPT}(G)/4$, and (ii) $L \geq p \cdot \text{OPT}(G)/4$. For case (i),

$$\begin{aligned}\Pr(\mathsf{E}_{win}) &\geq \Pr\left(\widehat{\text{OPT}} > 2L\right) \cdot \Pr\left(\mathsf{E}_{win} \mid \widehat{\text{OPT}} > 2L\right) \\ &\geq \Pr\left(\widehat{\text{OPT}} > p \cdot \text{OPT}(G)/2\right) \cdot \Pr\left(\mathsf{E}_{win} \mid \widehat{\text{OPT}} > 2L\right) = (1 - o(1)) \cdot 1\end{aligned}$$

where the last equality is by Eq (4) and Lemma 5.3. For case (ii),

$$\begin{aligned}\Pr(\mathsf{E}_{win}) &= \Pr\left(\widehat{\text{OPT}} > 2L\right) \cdot \Pr\left(\mathsf{E}_{win} \mid \widehat{\text{OPT}} > 2L\right) + \Pr\left(\widehat{\text{OPT}} \leq 2L\right) \cdot \Pr\left(\mathsf{E}_{win} \mid \widehat{\text{OPT}} \leq 2L\right) \\ &= \Pr\left(\widehat{\text{OPT}} > 2L\right) + \Pr\left(\widehat{\text{OPT}} \leq 2L\right) \cdot \Pr\left(\mathsf{E}_{win} \mid \widehat{\text{OPT}} \leq 2L\right) \qquad \text{(By Lemma 5.3)} \\ &\geq \Pr\left(\widehat{\text{OPT}} > 2L\right) + \Pr\left(\widehat{\text{OPT}} \leq 2L\right) \cdot \Pr\left(\text{ALG} \geq (1-\varepsilon) \cdot L \mid \widehat{\text{OPT}} \leq 2L\right) \\ &\geq \Pr\left(\widehat{\text{OPT}} > 2L\right) \cdot \Pr\left(\text{ALG} \geq (1-\varepsilon) \cdot L \mid \widehat{\text{OPT}} > 2L\right) \\ &\qquad + \Pr\left(\widehat{\text{OPT}} \leq 2L\right) \cdot \Pr\left(\text{ALG} \geq (1-\varepsilon) \cdot L \mid \widehat{\text{OPT}} \leq 2L\right) \\ &= \Pr(\text{ALG} \geq (1-\varepsilon) \cdot L) = 1 - o(1)\end{aligned}$$

where the last equality is by Lemma 5.2. $\square$

Theorem 5.1 now follows from Lemma 5.4 (by replacing ε with $\varepsilon/2$ in Algorithm 3), and the fact that Algorithm 3 queries $O(\frac{\log(1/\varepsilon p)}{\varepsilon p})$ matchings and hence $O(\frac{\log(1/\varepsilon p)}{\varepsilon p})$ incident edges are queried for each vertex.

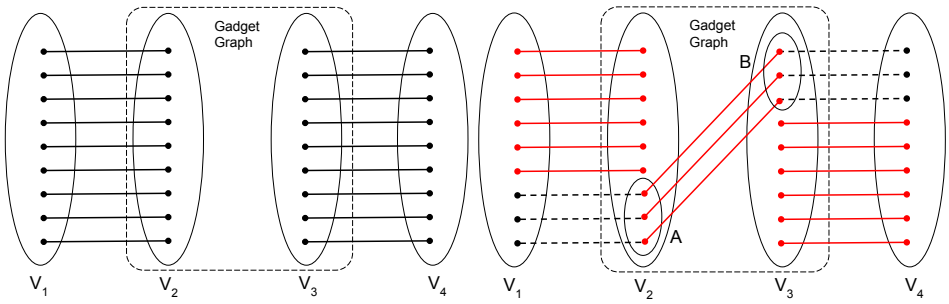
6. A BARRIER TO OBTAINING A NON-ADAPTIVE $(1-\varepsilon)$-APPROXIMATION ALGORITHM

The approximation ratio of our non-adaptive algorithm is $(\frac{1}{2} - \varepsilon)$ as opposed to the near-optimal ratio of $(1-\varepsilon)$ achieved by our adaptive algorithm. A natural question, first raised by [Blum et al. 2015], is if one can obtain a $(1-\varepsilon)$-approximation using a non-adaptive algorithm, even by allowing arbitrary dependence on p and ε. In the following, we highlight a possible barrier to obtain such a result.

Consider a bipartite graph $G(L, R, E)$ constructed as follows: (i) the vertex sets are $L = V_1 \cup V_3$, $R = V_2 \cup V_4$ and $|V_i| = N$ for $i \in [4]$, (ii) there is a perfect matching between V_1 and V_2, and a perfect matching between V_3 and V_4, and (iii) there is a *gadget graph* $\widehat{G}(V_2, V_3, \widehat{E})$, to be determined later, between V_2 and V_3 (see Fig 2.a).

Suppose we want to design a non-adaptive $(1-\varepsilon)$-approximation algorithm for the instance $G(V, E)$ with the parameter $p = 2/3$. In this case, for any graph G_p, w.h.p., there is a matching M_1 between V_1 and V_2, and another matching M_2 between V_3 and V_4, each of size $(2/3)N - o(N)$. Hence,

$$\text{OPT}(G_p) \geq |M_1| + |M_2| + (2/3) \cdot m(A, B) - o(N) \tag{5}$$



(a) Input graph $G(V, E)$ (b) A realization $G_p(V, E_p)$

Fig. 2: An example of a barrier to $(1-\varepsilon)$-approximation non-adaptive algorithms. The edges in the gadget graph are *not* presented in this figure. In part (b), solid red edges (resp. dashed edges) are the edges that are realized (resp. not realized).

where A (resp. B) is the set of vertices in V_2 (resp. V_3) that are not matched by M_1 (resp. M_2), and $m(A, B)$ denotes the size of a maximum matching between A and B. A few observations are in order. First, picking edges of M_1 and M_2 is crucial for having any large matching in G_p, and second, for a uniformly at random chosen realization of M_1 and M_2, the set A and B are chosen uniformly at random from V_2 and V_3 (see Fig 2.b). Based on these observations, we define the following problem.

PROBLEM 6.1. *Given a bipartite graph $G(L, R, E)$, choose a subgraph $H(L, R, Q)$ such that given two subsets $A \subseteq L$ and $B \subseteq R$, if $m(A, B) \geq N/3 - o(N)$ in G, then H contains* at least $\Omega(N)$ *edges between A and B.*

The goal is to solve Problem 6.1 using a graph H with small number of edges. The previous discussion implies that any non-adaptive $(1-\varepsilon)$-approximation algorithm has to solve Problem 6.1 for the gadget graph $\widehat{G}$, when the two sets A and B are chosen *uniformly at random*. Otherwise, for the maximum matching size $\text{ALG}(G_p)$ in $H(V, Q_p)$ and maximum matching size $\text{OPT}(G_p)$ in G_p, we have:

$$\begin{aligned} &\text{ALG}(G_p) \leq |M_1| + |M_2| + o(N) \leq (4/3)N + o(N) \\ &\text{OPT}(G_p) \geq (4/3)N + (2/3)(N/3) - o(N) = (14/9)N - o(N) \qquad \text{(by Eq (5))} \end{aligned}$$

Hence, the approximation ratio of the algorithm on this instance is at most $6/7 + o(1)$, bounded away from being a $(1-\varepsilon)$ approximation for $\varepsilon < 1/7$.

Although for randomly chosen subsets A and B, no lower bound on the size of H is known, we show in the following that if A and B are chosen *adversarially*, then there exist graphs for which solving Problem 6.1 requires storing a subgraph with *super linear* in n number of edges. Note that the number of queries of any non-adaptive algorithm is at least the number of edges in H and hence this bound on the number of edges in H implies that $\omega(n)$ queries are needed, or in other words, the number of per-vertex query needs to be a *function of* n. The existence of such a graph indicates a barrier to obtain a non-adaptive $(1-\varepsilon)$-approximation algorithm.

To continue, we need a few definitions. For a graph $G(V, E)$, a matching M is called an *induced matching*, if there is no edge between the vertices matched in M, i.e., $V(M)$, except for the edges in M. A graph $G(V, E)$ is called an (r, t)-*Ruzsa-Szemerédi* graph ((r, t)-RS graph for short), if the edge set E can be partitioned into t *induced matchings* each of size r. Note than the number of edges in any (r, t)-RS graph is $r \cdot t$.

Suppose $G(L, R, E)$ is an (r, t)-RS graph with the parameter $r = N/3$ and induced matchings $M_1, \ldots, M_t$. For each $i \in [t]$, define A_i (resp. B_i) as $V(M_i) \cap L$ (resp. $V(M_i) \cap R$). Suppose we choose the pair (A, B) only from the set of pairs $\mathcal{F} := \{(A_1, B_1), \ldots, (A_t, B_t)\}$. Note that between any pair in $\mathcal{F}$, there is a matching of size $r = N/3$, and moreover all edges of G are partitioned between these matchings. If the subgraph $H(V, Q)$ has only $o(r \cdot t)$ edges, a simple counting argument suggests that for $1 - o(1)$ fraction of pairs in $\mathcal{F}$, only $o(r)$ edges between the pairs are present in H. Hence, H cannot be a solution to Problem 6.1.

To complete the argument, we point out that there are (r, t)-RS graphs on $2N$ vertices with parameters $r = N/3$ and $t = N^{\Omega(1/\log\log N)}$ [Fischer et al. 2002; Goel et al. 2012]. These constructions certify that to solve Problem 6.1 when the sets A and B are chosen adversarially, one needs to store a subgraph with $n^{1+\Omega(1/\log\log n)} = \omega(n \cdot \text{polylog}(n))$ edges. In conclusion, while this result does not rule out the possibility of a non-adaptive $(1-\varepsilon)$-approximation algorithm where the number of per-vertex queries is independent of n, it suggests that any such algorithm has to crucially overcome Problem 6.1 using the fact that the two sets A and B are chosen randomly instead of adversarially.

7. CONCLUSIONS

We studied the stochastic matching problem in this paper. We showed that there exists an adaptive $(1-\varepsilon)$-approximation algorithm for this problem with $O(\frac{\log(1/\varepsilon p)}{\varepsilon p})$ per-vertex queries and degree of adaptivity. We further presented a non-adaptive $(\frac{1}{2} - \varepsilon)$-approximation algorithm with $O(\frac{\log(1/\varepsilon p)}{\varepsilon p})$ per-vertex queries. These results represent an exponential improvement over the previous best bounds of [Blum et al. 2015], answering an open problem in that work.

An interesting direction for future research is to design a non-adaptive algorithm that obtains a better than $\frac{1}{2}$-approximation while maintaining the property that the number of per-vertex queries is independent of n. Toward this direction, we highlighted a potential barrier to achieve a $(1-\varepsilon)$-approximation non-adaptively.

ACKNOWLEDGMENTS

We would like to thank Nika Haghtalab for introducing us to the stochastic matching problem. This work was supported in part by National Science Foundation grants CCF-1116961, CCF-1552909, and IIS-1447470.

REFERENCES

Marek Adamczyk. 2011. Improved analysis of the greedy algorithm for stochastic matching. *Inf. Process. Lett.* 111, 15 (2011), 731–737.

Mohammad Akbarpour, Shengwu Li, and Shayan Oveis Gharan. 2014. Dynamic matching market design. In *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*. 355.

Ross Anderson, Itai Ashlagi, David Gamarnik, and Yash Kanoria. 2015a. A dynamic model of barter exchange. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*. 1925–1933.

Ross Anderson, Itai Ashlagi, David Gamarnik, and Alvin E. Roth. 2015b. Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences* 112, 3 (2015), 663–668.

Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. 2016. Maximum Matchings in Dynamic Graph Streams and the Simultaneous Communication Model. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. 1345–1364.

Pranjal Awasthi and Tuomas Sandholm. 2009. Online Stochastic Optimization in the Large: Application to Kidney Exchange. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009*. 405–411.

Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. 2012. When LP Is the Cure for Your Matching Woes: Improved Bounds for Stochastic Matchings. *Algorithmica* 63, 4 (2012), 733–762.

Avrim Blum, John P. Dickerson, Nika Haghtalab, Ariel D. Procaccia, Tuomas Sandholm, and Ankit Sharma. 2015. Ignorance is Almost Bliss: Near-Optimal Stochastic Matching With Few Queries. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*. 325–342.

Avrim Blum, Anupam Gupta, Ariel D. Procaccia, and Ankit Sharma. 2013. Harnessing the power of two crossmatches. In *ACM Conference on Electronic Commerce, EC '13, Philadelphia, PA, USA, June 16-20, 2013*. 123–140.

Ning Chen, Nicole Immorlica, Anna R. Karlin, Mohammad Mahdian, and Atri Rudra. 2009. Approximating Matches Made in Heaven. In *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Part I*. 266–278.

Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. 2016. Kernelization via Sampling with Applications to Finding Matchings and Related Problems in Dynamic Graph Streams. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*. 1326–1344.

Kevin P. Costello, Prasad Tetali, and Pushkar Tripathi. 2012. Stochastic Matching with Commitment. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Proceedings, Part I*. 822–833.

John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. 2012. Dynamic Matching via Weighted Myopia with Application to Kidney Exchange. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. 2013. Failure-aware kidney exchange. In *ACM Conference on Electronic Commerce, EC '13*. 323–340.

John P. Dickerson and Tuomas Sandholm. 2015. FutureMatch: Combining Human Value Judgments and Machine Learning to Match in Dynamic Environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 622–628.

Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. 2002. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing*. 474–483.

Ashish Goel, Michael Kapralov, and Sanjeev Khanna. 2012. On the Communication and Streaming Complexity of Maximum Bipartite Matching. In *the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '12)*. SIAM, 468–485.

Anupam Gupta and Viswanath Nagarajan. 2013. A Stochastic Probing Problem with Applications. In *Integer Programming and Combinatorial Optimization - 16th International Conference, IPCO 2013. Proceedings*. 205–216.

L. Lovász and D. Plummer. 2009. *Matching Theory*. American Mathematical Soc. https://books.google.com/books?id=yW3WSVq8ygcC

David F. Manlove and Gregg O'Malley. 2014. Paired and Altruistic Kidney Donation in the UK: Algorithms and Experimentation. *ACM Journal of Experimental Algorithmics* 19, 1 (2014).

Alessandro Panconesi and Aravind Srinivasan. 1997. Randomized Distributed Edge Coloring via an Extension of the Chernoff-Hoeffding Bounds. *SIAM J. Comput.* 26, 2 (1997), 350–368.

Utku Unver. 2010. Dynamic Kidney Exchange. *Review of Economic Studies* 77, 1 (2010), 372–414.