Approximating Matches Made in Heaven

Ning Chen[1], Nicole Immorlica[2], Anna R. Karlin[3], Mohammad Mahdian[4], and Atri Rudra[5]

[1] Nanyang Technological University, Singapore.
[2] Northwestern University, Chicago, IL, U.S.A.
[3] University of Washington, Seattle, WA, U.S.A.
[4] Yahoo! Research, Santa Clara, CA, U.S.A.
[5] University at Buffalo, State University of New York, Buffalo, NY, U.S.A.

Abstract. Motivated by applications in online dating and kidney exchange, we study a stochastic matching problem in which we have a random graph G given by a node set V and probabilities $p(i, j)$ on all pairs $i, j \in V$ representing the probability that edge (i, j) exists. Additionally, each node has an integer weight $t(i)$ called its patience parameter. Nodes represent agents in a matching market with dichotomous preferences, i.e., each agent finds every other agent either *acceptable* or *unacceptable* and is indifferent between all acceptable agents. The goal is to maximize the welfare, or produce a matching between acceptable agents of maximum size. Preferences must be solicited based on probabilistic information represented by $p(i, j)$, and agent i can be asked at most $t(i)$ questions regarding his or her preferences.

A stochastic matching algorithm iteratively probes pairs of nodes i and j with positive patience parameters. With probability $p(i, j)$, an edge exists and the nodes are irrevocably matched. With probability $1 - p(i, j)$, the edge does not exist and the patience parameters of the nodes are decremented. We give a simple greedy strategy for selecting probes which produces a matching whose cardinality is, in expectation, at least a quarter of the size of this optimal algorithm's matching. We additionally show that variants of our algorithm (and our analysis) can handle more complicated constraints, such as a limit on the maximum number of rounds, or the number of pairs probed in each round.

1 Introduction

Matching is a fundamental primitive of many markets including job markets, commercial markets, and even dating markets [3–5, 16–18]. While matching is a well understood graph-theoretic concept, its stochastic variants are considerably less well-developed. Yet stochastic variants are precisely the relevant framework for most markets which incorporate a degree of uncertainty regarding the preferences of the agents. In this paper we study a stochastic variant of matching motivated by applications in the kidney exchange and online dating markets, or more generally, for matching markets with dichotomous preferences in which each agent finds every other agent either *acceptable* or *unacceptable* and is indifferent between acceptable agents (see, e.g., [6]). The basic stochastic matching problem, which is the main focus of this paper, can be stated as follows:

> Let G be a random undirected graph given by a node set V (representing agents in the matching market) and a probability $p(i,j)$ on any pair i,j of nodes, representing the probability that an edge exists between that pair of nodes (i.e., the probability that the corresponding agents find each other acceptable). Whether or not there is an edge between a pair of nodes is not revealed to us unless we *probe* this pair (solicit the preference information from the relevant agents). Upon probing a pair, if there is an edge between them, they are matched and removed from the graph. In other words, when a pair (i,j) is probed, a coin is flipped with probability $p(i,j)$. Upon heads, the pair is matched and leaves the system. Also, for every node i, we are given a number $t(i)$ called the *patience parameter* of i, which specifies the maximum number of failed probes i is willing to participate in.
>
> The goal is to maximize the welfare, i.e., design a probing strategy to maximize the expected number of matches.

The above formulation of the problem is similar in nature to the formulation of other stochastic optimization problems such as stochastic shortest path [12, 7] and stochastic knapsack [8]. The stochastic matching problem is an exponential-sized Markov Decision Process (MDP) and hence has an optimal dynamic program, also exponential. Our goal is to approximate the expected value of this dynamic program in polynomial time. We show that a simple non-adaptive greedy algorithm that runs in near-linear time is a 4-approximation (Section 3). The algorithm simply probes edges in order of decreasing probability. Our algorithm is practical, intuitive, and near-optimal. Interestingly, the algorithm need not even know the patience parameters, but just which edges are more probable.

It is easy to see that the above greedy algorithm is a good approximation when the patience parameters are all one or all infinite: when the patience parameters are all one, the optimal algorithm clearly selects a maximum matching and so the maximal matching selected by the greedy algorithm is a 2-approximation; when the patience parameters are all infinite, for any instantiation of the coin flips, the greedy algorithm finds a maximal matching and hence is a 2-approximation to the (ex-post) maximum matching. To prove that the greedy algorithm is a constant approximation in general, we can no longer compare our performance to the expected size of the maximum matching. As we show in Appendix A, the gap between the expected size of the maximum matching and the expected value of the optimum algorithm may be larger than any constant. Instead, we compare the decision tree of the greedy algorithm to the decision tree of the optimum algorithm. Using induction on the graph as well as a careful charging scheme, we are able to show that the greedy algorithm is a 4-approximation for general patience parameters. Unfortunately, we do not know if computing the optimal solution is even NP-hard. Further, we do not know whether if the analysis of the greedy algorithm is tight. We leave these as open questions and conjecture that (i) computing the optimal strategy is indeed NP-hard and (ii) the greedy algorithm is indeed a 2-approximation.

We also show that our algorithm and analysis can be adapted to handle more complicated constraints (Section 4). In particular, if probes must be performed in a limited number of rounds, each round consisting of probing a matching, a natural generalization of the greedy algorithm gives a 6-approximation in the uniform probability case. For this generalization, the problem does turn out to be NP-hard (Appendix B). We can also generalize the algorithm to a case where we only probe a limited number of edges in each round (Section 4).

1.1 Motivation

In addition to being an innately appealing and natural problem, the stochastic matching problem has important applications. We outline here two applications to kidney exchange and online dating.

Kidney Exchange. Currently, there are 98,167 people in need of an organ in the United States. Of these, 74,047 patients are waiting for a kidney.[6] Every healthy person has two kidneys, and only needs one kidney to survive. Hence it is possible for a living friend or family of the patient to donate a kidney to the patient. Unfortunately, not all patients have compatible donors. At the recommendation of the medical community [14, 15], in year 2000 the United Network for Organ Sharing (UNOS) began performing *kidney exchanges* in which two incompatible patient/donor pairs are identified such that each donor is compatible with the other pair's patient. Four simultaneous operations are then performed, exchanging the kidneys between the pairs in order to have two successful transplants.

To maximize the total number of kidney transplants in the kidney exchange program, it is important to match the maximum number of pairs. This problem can be phrased as that of maximum matching on graphs in which the nodes represent incompatible pairs and the edges represent possible transplants based on medical tests [17, 18]. There are three main tests which indicate the likelihood of successful transplants. The first two tests, the blood-type test and the antibody screen, compare the blood of the recipient and donor. The third test, called *crossmatching* combines the recipient's blood serum with some of the donor's red blood cells and checks to see if the antibodies in the serum kill the cells. If this happens (the crossmatch is *positive*), then the transplant can not be performed. If this doesn't happen (the crossmatch is *negative*), then the transplant may be performed.[7]

Of course, the feasibility of a transplant can only be determined after the final crossmatch test. As this test is time-consuming and must be performed close to the surgery date [2, 1], it is infeasible to perform crossmatch tests on all nodes in the graph. Furthermore, due to incentives facing doctors, it is important to perform a transplant as soon as a pair with negative crossmatch tests is identified. Thus the edges are really stochastic; they only reflect the *probability*, based on

---

[6] Data retrieved on November 19th, 2007 from United Network for Organ Sharing (UNOS) — The Organ Procurement and Transplantation Network (OPTN), `http://www.optn.org/data`.

[7] Recent advances in medicine actually allow positive crossmatch transplants as well, but these are significantly more risky.

the initial two tests and related factors, that an exchange is possible. Based on this information alone, edges must be selected and, upon a negative crossmatch test, the surgery performed. Hence the matching problem is actually a stochastic matching problem. The patience parameters in the stochastic matching problem can be used to model the unfortunate fact that patients will eventually die without a successful match.

Online Dating. Another relevant marketplace for stochastic matching is the online dating scene, the second-largest paid-content industry on the web, expected to gross around $600 million in 2008 [9]. In many online dating sites, most notably eHarmony and Just Lunch, users submit profiles to a central server. The server then estimates the compatibility of a couple and sends plausibly compatible couples on blind dates (and even virtual blind dates). The purported goal of these sites is to create as many happily married couples as possible.

Again, this problem may be modeled as a stochastic matching problem. Here, the people participating in the online match-making program are the nodes in the graph. From the personal characteristics of these individuals, the system deduces for each pair a probability that they are a good match. Whether or not a pair is actually successful can only be known if they are sent on a date. In this case, if the pair is a match, they will immediately leave the program. Also, each person is willing to participate in at most a given number of unsuccessful dates before he/she runs out of patience and leaves the match-making program. The online dating problem is to design a schedule for dates to maximize the expected number of matched couples.

2 Preliminaries

The stochastic matching problem can be represented by a random graph $G = (V, E)$, where for each pair (α, β) of vertices, there is an undirected edge between α and β with a probability $p(\alpha, \beta) \in [0, 1]$.[8] For the rest of the paper, w.l.o.g. we will assume that E contains exactly the pairs that have positive probability. These probabilities are all independent. Additionally, for each vertex $\gamma \in V$ a number $t(\gamma)$ called the *patience parameter* of γ is given. The existence of an edge between a pair of vertices of the graph is only revealed to us after we *probe* this pair. When a pair (α, β) is probed, a coin is flipped with probability $p(\alpha, \beta)$. Upon heads, the pair is matched and is removed from the graph. Upon tails, the patience parameter of both α and β are decremented by one. If the patience parameter of a node reaches 0, this node is removed from the graph. This guarantees that each vertex γ can be probed at most $t(\gamma)$ times. The problem is to design (possibly adaptive) strategies to probe pairs of vertices in the graph such that the expected number of matched pairs is maximized.

An instance of our problem is thus a tuple (G, t). For a given algorithm $\mathtt{ALG}$, let $\mathbf{E}_{\mathtt{ALG}}(G, t)$ (or $\mathbf{E}_{\mathtt{ALG}}(G)$ for simplicity, when t is clear from the context) be

[8] Note that here we do not impose any constraint that the graph G should be bipartite. In settings such as heterosexual dating where such a constraint is natural, it can be imposed by setting the probabilities between vertices on the same side to zero.

the expected number of pairs matched by ALG, where the expectation is over the realizations of probes and (possible) coin tosses of the algorithm itself.

Decision Tree Representation. For any deterministic algorithm ALG and any instance (G, t) of the problem, the entire operation of ALG on (G, t) can be represented as an (exponential-sized) *decision tree* $T_{\texttt{ALG}}$. The root of $T_{\texttt{ALG}}$, r, represents the first pair $e = (\alpha, \beta) \in E$ probed by ALG. The *left* and the *right* subtrees of r represent *success* and *failure* for the probe to (α, β), respectively. In general, each node of this tree corresponds to a probe and the left and the right subtrees correspond to the respective success or failure.

For each node $v \in T_{\texttt{ALG}}$, a corresponding sub-instance (G_v, t_v) of the problem can be defined recursively as follows: The root r corresponds to the initial instance (G, t). If a node v that represents a probe to a pair (α, β) corresponds to (G_v, t_v),

- the left child of v corresponds to $(G_v \setminus \{\alpha, \beta\}, t_v)$, and
- the right child of v corresponds to $(G_v \setminus \{(\alpha, \beta)\}, t'_v)$, where $G_v \setminus \{(\alpha, \beta)\}$ denotes the instance obtained from G_v by setting the probability of the edge (α, β) to zero, and $t'_v(\alpha) = t_v(\alpha) - 1$, $t'_v(\beta) = t_v(\beta) - 1$ and $t'_v(\gamma) = t_v(\gamma)$ for any other vertex γ.

For each node $v \in T_{\texttt{ALG}}$, let T_v be the subtree rooted at v. Let $T_{L(v)}$ and $T_{R(v)}$ be the left and right subtree of v, respectively. Observe that T_v essentially defines an algorithm ALG′ on the sub-instance (G_v, t_v) corresponding to v. Define $\mathbf{E}_{\texttt{ALG}}(T_v)$ to be the expected value generated by the algorithm corresponding to ALG′, i.e. $\mathbf{E}_{\texttt{ALG}}(T_v) = \mathbf{E}_{\texttt{ALG}'}(G_v, t_v)$.

The stochastic matching problem can be viewed as the problem of computing the optimal policy in an exponential-sized Markov Decision Process (for more details on MDPs, see the textbook by Puterman [13]). The states of this MDP correspond to subgraphs of G that are already probed, and the outcome of these probes. The actions that can be taken at a given state correspond to the choice of the next pair to be probed. Given an action, the state transitions probabilistically to one of two possible states, one corresponding to a success, and the other corresponding to a failure in the probe. We denote by OPT the optimal algorithm, i.e., the solution of this MDP. Note that we can assume without loss of generality that OPT is deterministic, and therefore, a decision tree $T_{\texttt{OPT}}$ representing OPT can be defined as described above. Observe that by definition, for any node v of this tree, if the probability of reaching v from the root is non-zero, the algorithm defined by T_v must be the optimal for the instance (G_v, t_v) corresponding to v. To simplify our arguments, we assume without loss of generality that the algorithm defined by T_v is optimal for (G_v, t_v) for *every* $v \in T_{\texttt{OPT}}$, even for nodes v that have probability zero of being reached. Note that such nodes can exist in $T_{\texttt{OPT}}$, since OPT can probe edges of probability 1, in which case the corresponding right subtree is never reached.

Note that it is not even clear that the optimal strategy OPT can be described in polynomial space. Therefore, one might hope to use other benchmarks such as the optimal offline solution (i.e., the expected size of maximum matching in

G) as an upper bound on OPT. However, as we show in Appendix A, the gap between OPT and the optimal offline solution can be larger than any constant.

3 Greedy Algorithm

We consider the following greedy algorithm.

GREEDY.
1. Sort all edges in E by probabilities, say, $p(e_1) \geq p(e_2) \geq \cdots \geq p(e_m)$
   (ties are broken arbitrarily)
2. For $i = 1, \ldots, m$
3. if the two endpoints of e_i are available, probe e_i

Our main result is as follows.

Theorem 1 *For any instance graph (G, t), GREEDY is a 4-approximation to the optimal algorithm, i.e.* $\mathbf{E}_{\mathtt{OPT}}(G, t) \leq 4 \cdot \mathbf{E}_{\mathtt{GREEDY}}(G, t)$.

In the rest of this section, we will prove Theorem 1. The proof is inductive and is based on carefully charging the value obtained at different nodes of $T_{\mathtt{OPT}}$ to $T_{\mathtt{ALG}}$. We will begin by establishing two lemmas that will be useful for the proof. (The proofs are deferred to Appendix C.)

Lemma 1. *For any node $v \in T_{\mathtt{OPT}}$,* $\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right) \leq \mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right) \leq 1 + \mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)$.

Lemma 2. *For any node $v \in T_{\mathtt{OPT}}$, assume v represents the edge $e = (\alpha, \beta) \in E$, and let $p = p(\alpha, \beta)$ be the probability of e. If we increase the probability of v to $p' > p$ in $T_{\mathtt{OPT}}$, then $\mathbf{E}_{\mathtt{OPT}}(T_{\mathtt{OPT}})$ will not decrease.*

Note that Lemma 2 does not mean we increase the probability of edge e in graph G. It only says for a particular probe of e in $T_{\mathtt{OPT}}$, which corresponds to node v in the claim, if the probability of e is increased, the expected value of OPT will not decrease.

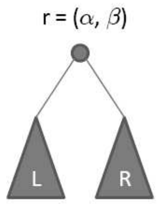

Fig. 1. Greedy tree $T_{\mathtt{GREEDY}}$

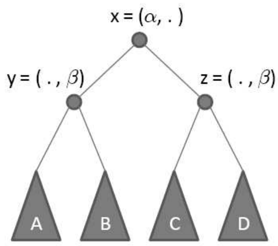

Fig. 2. Optimum tree $T_{\mathtt{OPT}}$

These two lemmas provide the key ingredients of our proof. To get an idea of the proof, imagine that the first probe of the greedy algorithm is to edge (α, β) represented by node r at the root of $T_{\mathtt{GREEDY}}$ as in Figure 1 and suppose that $T_{\mathtt{OPT}}$ is as in Figure 2. Let p_r be the probability of success of probe (α, β).

Note the algorithm ALG_1 defined by subtree A in $T_{\texttt{OPT}}$ is a valid algorithm for the left subtree of greedy (since the optimum algorithm has already matched nodes α and β upon reaching subtree A, all probes in subtree A are valid probes for the left-subtree of $T_{\texttt{GREEDY}}$). Furthermore, ALG_1 achieves the same value, in expectation, as the optimum algorithm on subtree A. Similarly the algorithm ALG_2 defined by subtree D in $T_{\texttt{OPT}}$ is a valid algorithm for the right subtree of greedy *except* ALG_2 may perform a probe to (α, β). Thus we define a secondary (randomized) algorithm ALG_2' which follows ALG_2 but upon reaching a probe to (α, β) simply flips a coin with probability p_r to decide which subtree to follow and does not probe the edge. Hence ALG_2' is a valid algorithm for the right subtree of greedy, and gets the same value as the optimum algorithm on subtree D minus a penalty of p_r for the missed probe to (α, β). The value of ALG_1 and ALG_2' on the left and right subtree of $T_{\texttt{GREEDY}}$ respectively is at most the value of the optimum algorithm on those subtrees and so, by the inductive hypothesis, at most four times the value of the greedy algorithm on those subtrees. By Lemma 2, we can assume the probes at nodes x, y, and z in $T_{\texttt{OPT}}$ have probability p_r of success. Furthermore, we can use Lemma 1 to bound the value of the optimum algorithm in terms of the left-most subtree A and the right-most subtree D. With a slight abuse of notation, we use A to denote the expected value of the optimum algorithm on subtree A (and similarly, B, C, and D). Summarizing the above observations, we then get:

$$\begin{aligned}
\mathbf{E}_{\texttt{OPT}}(G,t) &\leq p_r^2(A+2) + p_r(1-p_r)(B+1) + p_r(1-p_r)(C+1) + (1-p_r)^2 D \\
&= 2p_r + p_r^2 A + p_r(1-p_r)B + p_r(1-p_r)C + (1-p_r)^2 D \\
&\leq 2p_r + p_r^2 A + p_r(1-p_r)(A+1) + p_r(1-p_r)D + (1-p_r)^2 D \\
&= 3p_r - p_r^2 + p_r A + (1-p_r)D \\
&\leq 4p_r + p_r A + (1-p_r)(D-p_r) \\
&= 4 \cdot \left(p_r(1+\mathbf{E}_{\texttt{ALG}_1}) + (1-p_r)\mathbf{E}_{\texttt{ALG}_2'}\right) \\
&\leq 4\mathbf{E}_{\texttt{GREEDY}}(G,t)
\end{aligned}$$

where the first inequality is by Lemma 2, the second inequality is by Lemma 1, and the last inequality is by the inductive hypothesis.

The above sketch represents the crux of the proof. To formalize the argument, we must account for all possibilities of $T_{\texttt{OPT}}$. We do this by considering "frontiers" in $T_{\texttt{OPT}}$ representing initial probes to α and β, and then follow the general accounting scheme suggested above via slightly more complicated algebraic manipulations.

Proof of Theorem 1. The proof is by induction on the set of edges in the graph G and the patience parameters. In particular, (G', t') is a sub-instance of (G, t) if G' is an edge subgraph of G and for every vertex $v \in V(G')$, $t'(v) \leq t(v)$. In the base case where the graph has only one edge, the claim is obviously true. Assume that for any sub-instance (G', t') of instance (G, t),

$$\mathbf{E}_{\texttt{OPT}}(G', t') \leq 4 \cdot \mathbf{E}_{\texttt{GREEDY}}(G', t').$$

Given the induction hypothesis, we will show $\mathbf{E}_{\texttt{OPT}}(G,t) \le 4 \cdot \mathbf{E}_{\texttt{GREEDY}}(G,t)$.

Let r be the root of $T_{\texttt{GREEDY}}$, which represents probing the edge $(\alpha,\beta) \in E$, and p_r be the probability of edge (α,β). Let (G_L, t_L) and (G_R, t_R) be the subinstances corresponding to the left and right child of r, respectively. Note that

$$\mathbf{E}_{\texttt{GREEDY}}(G,t) = p_r + p_r \cdot \mathbf{E}_{\texttt{GREEDY}}(G_L, t_L) + (1-p_r) \cdot \mathbf{E}_{\texttt{GREEDY}}(G_R, t_R). \quad (1)$$

We consider two cases based on whether $p_r = 1$ or $p_r < 1$. If $p_r = 1$, then it is easy to see that the inductive hypothesis holds. Namely, let (G', t') be the subinstance of (G,t) obtained by removing edge (α,β). Then,

$$\mathbf{E}_{\texttt{OPT}}(G,t) \le \mathbf{E}_{\texttt{OPT}}(G',t') + 1 \le 4 \cdot \mathbf{E}_{\texttt{GREEDY}}(G',t') + 1 \le 4 \cdot \mathbf{E}_{\texttt{GREEDY}}(G,t),$$

where the second inequality follows from the inductive hypothesis.

If $p_r < 1$, then for every node $v \in T_{\texttt{OPT}}$, the probability p_v of the edge corresponding to v satisfies $0 < p_v < 1$ by the definition of the greedy algorithm:

We define q_v to be the probability that $\texttt{OPT}$ reaches node v in $T_{\texttt{OPT}}$. That is, q_v is the product of probabilities of all edges on the path from the root of $T_{\texttt{OPT}}$ to v. The following equality follows from the definition of q_v:

$$\mathbf{E}_{\texttt{OPT}}(G,t) = \sum_{v \in T_{\texttt{OPT}}} p_v \cdot q_v. \quad (2)$$

Define $X \subseteq T_{\texttt{OPT}}$ to be the set of nodes that correspond to the first time where $\texttt{OPT}$ probes an edge incident to α or β. In other words, X is the set of nodes $v \in T_{\texttt{OPT}}$ such that $\texttt{OPT}$ probes an edge incident to α or β (or both) at v and at none of the vertices on the path from the root to v (see figure 3 at the end of the paper for an example). Observe that no node in X lies in the subtree rooted at another node in X. Thus, X essentially defines a "frontier" in $T_{\texttt{OPT}}$.

Take a node $v \in X$. If v represents probing an edge incident to α, consider the set of all nodes in $T_{L(v)}$ that correspond to the *first* time an edge incident to β is probed; otherwise, consider all nodes in $T_{L(v)}$ that correspond to the first time an edge incident to α is probed. Let Y_1 be the union of all these sets, taken over all $v \in X$. Define $Y_2 \subseteq \bigcup_{v \in X} T_{R(v)}$ similarly, with $L(v)$ replaced by $R(v)$ (see Figure 3 at the end of the paper for an example).

For any subset of nodes $S \subseteq T_{\texttt{OPT}}$, define $T(S) = \bigcup_{v \in S} T_v$. In appendix C, we show that

$$\begin{aligned}\mathbf{E}_{\texttt{OPT}}(G,t) \le 3p_r &+ \sum_{v \in Y_1} q_v \cdot \mathbf{E}_{\texttt{OPT}}\left(T_{L(v)}\right) + \sum_{u \in X} \sum_{v \in T_{L(u)} \setminus T(Y_1)} p_v \cdot q_v \\ &+ \sum_{v \in Y_2} q_v \cdot \mathbf{E}_{\texttt{OPT}}\left(T_{R(v)}\right) + \sum_{u \in X} \sum_{v \in T_{R(u)} \setminus T(Y_2)} p_v \cdot q_v + \sum_{v \in T_{\texttt{OPT}} \setminus T(X)} p_v \cdot q_v \end{aligned} \quad (3)$$

Define an algorithm $\texttt{ALG}_1$ that works as follows: $\texttt{ALG}_1$ follows the decision tree of $\texttt{OPT}$ except that when the algorithm reaches a node $v \in X \cup Y_1$, it will not probe the edge corresponding to v and go to the left subtree $T_{L(v)}$ directly. Since in

$\mathtt{ALG}_1$ for every path from the root to a node in $\cup_{u\in T(Y_1)}$ (and $\cup_{u\in X} T_{L(u)} \setminus T(Y_1)$) has two (and one resp.) less successful probes in $X \cup Y_1$ than $\mathtt{OPT}$, it follows that

$$\begin{aligned}\mathbf{E}_{\mathtt{ALG}_1} &= \sum_{u\in X} \sum_{v\in Y_1\cap T_{L(u)}} \sum_{w\in T_{L(v)}} p_w \cdot \frac{q_w}{p_u p_v} + \sum_{u\in X} \sum_{w\in T_{L(u)}\setminus T(Y_1)} p_w \cdot \frac{q_w}{p_u} + \sum_{w\in T_{\mathtt{OPT}}\setminus T(X)} p_w \cdot q_w \\ &= \sum_{u\in X} \sum_{v\in Y_1\cap T_{L(u)}} \frac{q_v}{p_u} \cdot \mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right) + \sum_{u\in X} \sum_{w\in T_{L(u)}\setminus T(Y_1)} p_w \cdot \frac{q_w}{p_u} + \sum_{w\in T_{\mathtt{OPT}}\setminus T(X)} p_w \cdot q_w \quad (4)\end{aligned}$$

(recall $p_u > 0$, and hence the division is valid). In the second equality above, we have used the following fact. Fix $v \in Y_1$. For every $w \in T_{L(v)}$ let $q_w = q_v \cdot p_v \cdot q'_w$. Then $\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right) = \sum_{w\in T_{L(u)}} q'_w \cdot p_w$.

On the other hand, by the definition of X and Y_1, $\mathtt{ALG}_1$ will not probe any edge incident to α and β. Thus it is a valid algorithm for the instance (G_L, t_L). By the induction hypothesis, we have

$$\mathbf{E}_{\mathtt{ALG}_1} \le \mathbf{E}_{\mathtt{OPT}}(G_L, t_L) \le 4 \cdot \mathbf{E}_{\mathtt{GREEDY}}(G_L, t_L). \quad (5)$$

Define an algorithm $\mathtt{ALG}_2$ that works as follows: $\mathtt{ALG}_2$ follows the decision tree of $\mathtt{OPT}$ except that when the algorithm reaches a node $v \in X \cup Y_2$, it will not probe the edge corresponding to v and proceed to the right subtree $T_{R(v)}$ directly. Using an argument similar to the one used for $\mathbf{E}_{\mathtt{ALG}_1}$, we get

$$\begin{aligned}\mathbf{E}_{\mathtt{ALG}_2} &= \sum_{u\in X} \sum_{v\in Y_2\cap T_{R(u)}} \sum_{w\in T_{R(v)}} p_w \cdot \frac{q_w}{(1-p_u)(1-p_v)} + \sum_{u\in X} \sum_{w\in T_{R(u)}\setminus T(Y_2)} p_w \cdot \frac{q_w}{1-p_u} \\ &\quad + \sum_{w\in T_{\mathtt{OPT}}\setminus T(X)} p_w \cdot q_w \\ &= \sum_{u\in X} \sum_{v\in Y_2\cap T_{R(u)}} \frac{q_v}{1-p_u} \cdot \mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right) + \sum_{u\in X} \sum_{w\in T_{R(u)}\setminus T(Y_2)} p_w \cdot \frac{q_w}{1-p_u} \\ &\quad + \sum_{w\in T_{\mathtt{OPT}}\setminus T(X)} p_w \cdot q_w \quad (6)\end{aligned}$$

(recall $p_u < 1$, and hence the division is valid).

We define a variant $\mathtt{ALG}'_2$ from $\mathtt{ALG}_2$ where whenever $\mathtt{ALG}_2$ reaches a node corresponding to edge (α, β), $\mathtt{ALG}'_2$ will only make a coin toss with the same distribution to decide which subtree to go, but not probe the edge (α, β). I.e., the contribution of edge (α, β) is not included in $\mathtt{ALG}'_2$. It is easy to see that

$$\mathbf{E}_{\mathtt{ALG}_2} \le \mathbf{E}_{\mathtt{ALG}'_2} + p_r. \quad (7)$$

By the definition of X and Y_2, $\mathtt{ALG}'_2$ is a valid algorithm for the instance (G_R, t_R). By the induction hypothesis, we have

$$\mathbf{E}_{\mathtt{ALG}'_2} \le \mathbf{E}_{\mathtt{OPT}}(G_R, t_R) \le 4 \cdot \mathbf{E}_{\mathtt{GREEDY}}(G_R, t_R). \quad (8)$$

Now consider nodes $u \in X$ and imagine increasing the probability of success, p_u, to p_r for each such node. By Lemma 2, this can only increase the value of $\mathbf{E}_{\texttt{OPT}}(G,t)$ but it clearly does not change the value of ALG_1 or ALG_2. Let $\mathbf{E}_{\texttt{OPT}}(G,t)'$ be the value of the algorithm T_{OPT} on this new instance. From (3),

$$\begin{aligned}
\mathbf{E}_{\texttt{OPT}}(G,t)' \leq 3p_r &+ \sum_{u\in X}\sum_{v\in Y_1\cap T_{L(u)}} p_r\frac{q_v}{p_u}\cdot \mathbf{E}_{\texttt{OPT}}\left(T_{L(v)}\right) + \sum_{u\in X}\sum_{v\in T_{L(u)}\setminus T(Y_1)} p_v\cdot p_r\frac{q_v}{p_u} \\
&+ \sum_{u\in X}\sum_{v\in Y_2\cap T_{R(u)}} (1-p_r)\frac{q_v}{1-p_u}\cdot \mathbf{E}_{\texttt{OPT}}\left(T_{R(v)}\right) \\
&+ \sum_{u\in X}\sum_{v\in T_{R(u)}\setminus T(Y_2)} p_v\cdot(1-p_r)\frac{q_v}{1-p_u} + \sum_{v\in T_{\texttt{OPT}}\setminus T(X)} p_v\cdot q_v \\
&= 3p_r + p_r\cdot \mathbf{E}_{\texttt{ALG}_1} + (1-p_r)\cdot \mathbf{E}_{\texttt{ALG}_2}, \qquad (9)
\end{aligned}$$

where the last line follows from (4) and (6). Now consider the following sequence:

$$\begin{aligned}
\mathbf{E}_{\texttt{OPT}}(G,t) &\leq \mathbf{E}_{\texttt{OPT}}(G,t)' \leq 3p_r + p_r\cdot \mathbf{E}_{\texttt{ALG}_1} + (1-p_r)\cdot \mathbf{E}_{\texttt{ALG}_2} \\
&\leq 4p_r + p_r\cdot \mathbf{E}_{\texttt{ALG}_1} + (1-p_r)\cdot \mathbf{E}_{\texttt{ALG}'_2} \qquad (10) \\
&\leq 4p_r + 4p_r\cdot \mathbf{E}_{\texttt{GREEDY}}(G_L,t_L) + 4(1-p_r)\cdot \mathbf{E}_{\texttt{GREEDY}}(G_R,t_R) \qquad (11) \\
&= 4\cdot \mathbf{E}_{\texttt{GREEDY}}(G,t). \qquad (12)
\end{aligned}$$

In the above, (10) follows from (7), (11) follows from (5) and (8), and (12) follows from (1). This completes the proof.

4 Multiple Rounds Matching

In this section, we consider a generalization of the stochastic matching problem defined in Section 2. In this generalization, the algorithm proceeds in rounds, and is allowed to probe a set of edges (which have to be a matching) in each round. The additional constraint is a bound, k, on the maximum number of rounds. We show in Appendix B that finding the optimal strategy in this new model is NP-hard. Note that when k is large enough, the problem is equivalent to the model discussed in previous sections.

In the rest of this section, we will study approximation algorithms for the problem. By looking at the probabilities as the weights on edges, we have the following natural generalization of the greedy algorithm.

```
GREEDY_k.
1. For each round i = 1,...,k
2.    compute the maximum weighted matching in the current graph
3.    probe all edges in the matching
```

Let $\texttt{OPT}_k$ be the optimal algorithm under this setting. We would like to compare $\mathbf{E}_{\texttt{GREEDY}_k}$ against $\mathbf{E}_{\texttt{OPT}_k}$. Unfortunately, as the example in Appendix F shows, with no restriction on the instance, $\texttt{GREEDY}_k$ can be arbitrarily bad.

However, we can still prove that $\texttt{GREEDY}_k$ is a constant-factor approximation algorithm in two important special cases: when all nodes have infinite patience, and when nodes have arbitrary patience but all non-zero probability edges of

G have bounded probability (which contains the equal probability case). Furthermore, we observe that the latter result can be used to give a logarithmic approximation for the general case of the problem.

Special Cases. When the patience of all vertices are infinity, we will show that $\texttt{GREEDY}_k$ is a 4-approximation. (The proof is deferred to Appendix G.)

Theorem 2 *For any graph $G = (V, E)$, $\mathbf{E}_{\texttt{OPT}_k}[G] \leq 4 \cdot \mathbf{E}_{\texttt{GREEDY}_k}[G]$, when the patience of all vertices are infinity.*

Next, we study the approximability of $\texttt{GREEDY}_k$ on instances where nodes have arbitrary patience, but all edges of G have probabilities in a bounded range. (The proof is deferred to Appendix H.)

Theorem 3 *Let (G, t) be an instance such that for all pairs α, β of vertices, $p(\alpha, \beta)$ is either 0 or in $[p_{\min}, p_{\max}]$, for $0 < p_{\min} \leq p_{\max} \leq 1$. Then $\mathbf{E}_{\texttt{OPT}_k}[G] \leq (4 + 2p_{\max}/p_{\min}) \cdot \mathbf{E}_{\texttt{GREEDY}_k}[G]$.*

The General Case. Theorem 3 can be used to obtain a (randomized) approximation algorithm for the general case of the multi-round stochastic matching problem with an approximation factor of $O(\log n)$. This follows from the observations that one can delete all edges with probability less than $p_{\max}/n^2$ and the fact that Theorem 3 gives a constant factor approximation on subgraphs of G with edge weight in the range $(p_{max}/2^i, p_{\max}/2^{i+1}]$, for some integer $i \geq 0$. The details are in Appendix I.

A Further Extension. We also consider the following extension of the k-rounds model. In each round, an algorithm is only allowed to probe a matching of size at most C, where $1 \leq C \leq \lfloor |V|/2 \rfloor$ is another parameter (V is the set of vertices in the graph). Note that till now we have only considered the cases $C = 1$ and $C = \lfloor |V|/2 \rfloor$. Theorems 2 and 3 for the natural extension of the $\texttt{GREEDY}_k$ algorithm also hold in this model. Further, it can be shown that for the arbitrary patience and probability case, $\texttt{GREEDY}_k$ is a $\Theta(\min(k, C))$-approximation algorithm. See Appendix I for more details.

5 Conclusions

We studied natural greedy algorithms for the stochastic matching problem with patience parameters and proved that these algorithms are constant factor approximations. A natural question to ask is if designing the optimal strategy is computationally hard (this is even unknown for infinite patience parameters). In Appendix B we show the following two variants are NP-hard: (i) The algorithm can probe a matching in at most k rounds (the model we studied in Section 4) and (ii) the order in which the edges need to be probed are fixed (and the algorithm just needs to decide whether to probe an edge or not). In terms of positive results, it is well known that the greedy algorithm in Section 3 for the special cases of (i) all probabilities being 1 and (ii) all patience parameters being infinity is a 2-approximation. However, we proved that the greedy algorithm is a factor 4-approximation. We conjecture that the greedy algorithm is in fact a 2-approximation even for the general stochastic matching problem.

Another interesting variant of the problem is when edges also have weights associated with them and the objective is to maximize the (expected) total weight of the matched edges. In Appendix D we give an example that shows that the natural greedy algorithm has an unbounded approximation ratio. The greedy algorithm considered in Section 3 is *non-adaptive*, that is, the order of edges to probe are decided before the first probe. A natural question to ask is if there is a "gap" between the non-adaptive and *adaptive* optimal values? In Appendix E we show that the adaptive optimal is strictly larger than the non-adaptive optimal.

References

1. http://www.aakp.org/aakp-library/transplant-compatibility/index.cfm.
2. http://www.online-family-doctor.com/medical-tests/crossmatching.html.
3. A. Abdulkadiroglu, P.A. Pathak, and A.E. Roth. The new york city high school match. *American Economic Review*, 95(2):364–367, May 2005.
4. A. Abdulkadiroglu, P.A. Pathak, A.E. Roth, and T. Sonmez. The boston public school match. *American Economic Review*, 95(2):368–371, May 2005.
5. A. Abdulkadiroglu and T. Sonmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, June 2003.
6. A. Bogomolnaia and H. Moulin. Random matching under dichotomous preferences. *Econometrica*, 72(1):257–279, 2004.
7. M. Brand, J. Kelner, M. Mitzenmacher, and E. Nikolova. Stochastic shortest paths via quasi-convex maximization. In *Proceedings of 2006 European Symposium of Algorithms (ESA)*, 2006.
8. B. Dean, M. Goemans, and J. Vondrak. Approximating the stochastic knapsack: the benefit of adaptivity. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 208–217, 2004.
9. R. Epstein. The truth about online dating. *Scientific American*, February 2007.
10. P. Erdös and A. Réyni. On random matrices. *Publ. Math. Inst. Hungarian Academy of Sciences*, 8:455–461, 1964.
11. P. Erdös and A. Réyni. On random matrices II. *Studia Sci. Math. Hungar.*, 3:459–464, 1968.
12. C.H. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150, July 1991.
13. M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley Series in Probability and Statistics, 2005.
14. F.T. Rapaport. The case for a living emotionally related international kidney donor exchange registry. *Transplantation Proceedings*, XVIII:5–9, 1986.
15. L.F. Ross, D.T. Rubin, M. Siegler, M.A. Josephson, J.R. Thistlethwaite, and E.S. Woodle. Ethics of a paired-kidney-exchange program. *The New England Journal of Medicine*, CCCXXXVI:1752–1755, 1997.
16. A.E. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Economic Theory*, 92:991–1016, 1984.
17. A.E. Roth, T. Sonmez, and U. Unver. Kidney exchange. *Quarterly Journal of Economics*, 119(2):457–488, May 2004. Originally published as NBER Paper w10002, September 2003.
18. A.E. Roth, T. Sonmez, and U. Unver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151–188, December 2005. Originally published as NBER Paper w10698, August 2004.

A The Gap between OPT and the Optimal Offline Solution

In this appendix we give an example to show that the gap between the expected value of the optimal algorithm OPT and the expected value of the optimal algorithm that knows the realization of G (that is, the expected size of the maximum matching in G, denoted by $\mathbf{E}_{\texttt{MM}}[G]$) can be arbitrarily large. This means that one cannot hope to obtain an approximation algorithm for the stochastic matching problem using the optimal offline solution as a benchmark.

Proposition 1 *For every constant c, there exists a bipartite instance* (G, t) *of the stochastic matching problem such that* $\mathbf{E}_{\texttt{OPT}}(G) < c\mathbf{E}_{\texttt{MM}}[G]$.

Proof. Define G to be a complete bipartite graph with parts A and B each of size n and $p(\alpha, \beta) = \frac{2\log n}{n}$ for every pair $\alpha \in A$ and $\beta \in B$. By a result due to Erdös and Réyni [10, 11], it is known that with at least a constant probability, a random realization of G has a perfect matching. In other words, $\mathbf{E}_{\texttt{MM}}[G] = \Omega(n)$.

It is easy to see that for any algorithm ALG, and any vertex $\alpha \in A$,

$$\mathbf{Pr}[\alpha \text{ is matched by } \texttt{ALG}] \leq \texttt{t}(\alpha) \cdot \frac{2\log \texttt{n}}{\texttt{n}}.$$

Therefore, if we let $t(\alpha) = o\left(\frac{n}{\log n}\right)$ for every α, by linearity of expectation, $\mathbf{E}_{\texttt{ALG}}[G] = o(n)$, which completes the proof.

Note that the above example relied heavily on the patience parameters being finite. This is not a coincidence, as a simple argument shows that when the patience parameters are infinite, the gap between OPT and the expected size of the maximum matching is at most 2. We do not know if this bound is tight, but the following example gives a lower bound.

Example 1 *The following observation shows a gap even for the infinite patience case. Consider the* $G = K_{2,2}$ *graph with a probability of* $\frac{1}{2}$ *on every edge. The following straightforward calculations show that* $\mathbf{E}_{\texttt{MM}}[G] = \frac{22}{16}$, *while* $\mathbf{E}_{\texttt{OPT}}[G] = \frac{21}{16}$.

Note that all possible realizations occur with the same probability (and equal to $\frac{1}{16}$*). Now, in* 7 *realizations, the size of the maximum matching is* 2. *Thus, we have:*

$$\mathbf{E}_{\texttt{MM}}[G] = 2 \cdot \frac{7}{16} + 1 \cdot \frac{15-7}{16} = \frac{22}{16}.$$

Now to calculate the optimal value, first observe that the any edge can be probed first. After that we are left with a path of length three, for which we use the fact that the optimal algorithm always probes one of the "outer" edges. Thus, the optimal value is:

$$\frac{1}{2}\left(1 + \frac{1}{2}\right) + \frac{1}{2}\left(\frac{1}{2}\left(1 + \frac{1}{2}\right) + \frac{1}{2}\left(\frac{3}{4}\right)\right) = \frac{3}{4} + \frac{3}{8} + \frac{3}{16} = \frac{21}{16},$$

where in the above we have used the fact that the optimal values on paths of lengths two and one are $\frac{3}{4}$ *and* $\frac{1}{2}$ *respectively.*

In fact, by setting the probability of edges to a number $p \approx 0.7$ *instead of* $1/2$, *one can improve this example to get a gap of* ≈ 1.06.

B Hardness

In this appendix, we show that two more restricted versions of the stochastic matching problem are NP-hard. The NP-hardness of the basic stochastic matching problem (with no limit on the number of rounds) remains an open question.

B.1 Hardness of the Multi-round Stochastic Matching Problem

In the multiple rounds setting, it is NP-hard to find an optimal strategy to probe edges, even when all patience are infinite. Formally,

Theorem 4 *Finding the optimal algorithm for the multiple rounds matching problem is NP-hard.*

Proof. We reduce from the edge-coloring problem: Given a graph $G = (V, E)$, where $m = |E|$, we are asked if the edge set E is k-colorable (i.e. all edges incident to a same vertex have different colors).

The reduction is straightforward: We use the same graph G for the k-round matching problem, and for each edge $e \in E$, the probability that the edge is present is $\epsilon = 1/m^3$.

If graph G is k-colorable, then for each round $i = 1, \ldots, k$, we probe all edges with color $1, \ldots, k$, respectively. Note that the probe is feasible since edges have the same color form a matching. Thus, the expected number of matched edges is at least

$$\sum_{e \in E} \epsilon - \binom{m}{2} \epsilon^2 \geq m \cdot \frac{1}{m^3} - m^2 \cdot \frac{1}{m^6} = \frac{1}{m^2} - \frac{1}{m^4},$$

where the second term $\binom{m}{2}\epsilon^2$ is due to the following observation. For any pair of edges, with probability ϵ^2 both of them are present. However, we can probe only one edge when they are incident to a same vertex.

On the other hand, assume G is not k-colorable. For any algorithm, since in each round the algorithm probes a matching, we know at most $m - 1$ edges can be probed in k rounds. Thus, the expected number of matched edges is at most

$$(m-1)\epsilon = (m-1) \cdot \frac{1}{m^3} = \frac{1}{m^2} - \frac{1}{m^3} < \frac{1}{m^2} - \frac{1}{m^4}.$$

Therefore, if we can find the optimal algorithm in polynomial time, we are able to distinguish if the expected number of matched edges is at least $\frac{1}{m^2} - \frac{1}{m^4}$ or at most $\frac{1}{m^2} - \frac{1}{m^3}$, which is a contradiction.

B.2 A Hardness Result for a Given Order of Edges

Theorem 5 *For a graph G, assume we have to probe edges in a given order, say $e_1, \ldots, e_m$. That is, when it is the turn of edge e_i, we can either probe it or pass (in the latter case, we cannot probe e_i in the future). Then it is NP-hard to decide the best strategy to probe edges.*

Proof. (sketch) We reduce from the following restricted 3SAT problem: For a given formula $\phi(x_1, \ldots, x_n) = C_1 \wedge \cdots \wedge C_m$, where $m \geq n \geq 12$ and each literal x_i and $\bar{x}_i$ appear at most 4 times, we are asked if there is an assignment that satisfies all clauses.

We construct a graph G as follows: the vertex set is $\{a_i, u_i, v_i \mid i = 1, \ldots, n\} \cup \{b_j \mid j = 1, \ldots, m\}$, where u_i corresponds to $x_i = \mathit{false}$ and v_i corresponds to $x_i = \mathit{true}$, and edge set is

$$\{(a_i, u_i), (a_i, v_i) \mid i = 1, \ldots, n\} \cup \{(u_i, b_j) \mid C_j \text{ contains } x_i\} \cup \{(v_i, b_j) \mid C_j \text{ contains } \bar{x}_i\}.$$

The probability of edge (a_i, u_i) and (a_i, v_i) is 1, for $i = 1, \ldots, n$, and is $\epsilon = \frac{1}{m^2}$ for the rest of edges. The patience of vertex b_j is 1, for $j = 1, \ldots, m$, and is infinity of the rest of vertices. Now we assume the given order of edges is $(a_1, u_1), (a_1, v_1), \ldots, (a_n, u_n), (a_n, v_n), \ldots$.

If ϕ is satisfiable, then it can be seen that the expected number of matched edges is at least

$$n + m \cdot \epsilon - 2n\binom{4}{2}\epsilon^2 \geq n + \frac{1}{m} - 12m\frac{1}{m^4} = n + \frac{1}{m} - \frac{12}{m^3},$$

where the second term $2n\binom{4}{2}\epsilon^2$ means for each literal, with probability ϵ^2, we can probe only one edge.

On the other hand, if ϕ cannot be satisfied, then the expected number of matched edges is at most

$$n + (m-1)\epsilon = n + \frac{1}{m} - \frac{1}{m^2} \leq n + \frac{1}{m} - \frac{12}{m^3}.$$

Therefore, it is NP-hard to find the optimal strategy to probe edges in the given order.

C Missing Proofs from Section 3

C.1 Proof of Lemma 1

Let the node v in $T_{\texttt{OPT}}$ correspond to probing the edge $e = (\alpha, \beta) \in E$. Since $\texttt{OPT}$ reaches $T_{L(v)}$ if the probe to e succeeds and reaches $T_{R(v)}$ if the probe to e fails, $T_{L(v)}$ defines a valid algorithm on the instance $(G_{R(v)}, t_{R(v)})$ corresponding to $R(v)$. By the optimality of $\texttt{OPT}$ on every subtree, we have $\mathbf{E}_{\texttt{OPT}}\left(T_{L(v)}\right) \leq \mathbf{E}_{\texttt{OPT}}\left(T_{R(v)}\right)$.

On the other hand, since $T_{R(v)}$ is a valid algorithm for the sub-instance (G_v, t_v) corresponding to v,

$$\mathbf{E}_{\texttt{OPT}}\left(T_{R(v)}\right) \leq \mathbf{E}_{\texttt{OPT}}(T_v) = p(e) \cdot \left(1 + \mathbf{E}_{\texttt{OPT}}\left(T_{L(v)}\right)\right) + (1 - p(e)) \cdot \mathbf{E}_{\texttt{OPT}}\left(T_{R(v)}\right),$$

where the equality follows from the problem definition. The above implies that $\mathbf{E}_{\texttt{OPT}}(T_{R(v)}) \leq 1 + \mathbf{E}_{\texttt{OPT}}(T_{L(v)})$ as $p(e) > 0$.

C.2 Proof of Lemma 2

By Lemma 1 and the assumption that $p' > p$,

$$(p'-p)\mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right) \le (p'-p)\left(1+\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)\right),$$

which implies that

$$p\cdot\left(1+\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)\right)+(1-p)\cdot\mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right) \le p'+p'\cdot\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)+(1-p')\cdot\mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right).$$

The proof is complete by noting that the LHS and RHS of the above inequality corresponds to $\mathbf{E}_{\mathtt{OPT}}\left(T_v\right)$ before and after the probability of v is increased to p'.

C.3 Proof of Inequality (3)

We start with a couple of observations.

As X essentially defines a "frontier" in $T_{\mathtt{OPT}}$,

$$\sum_{v\in X} p_v\cdot q_v \le \max_{v\in X} p_v \le p_r, \tag{13}$$

where the last inequality follows from the definition of the greedy algorithm.

Also observe that $Y_1\cup Y_2$ defines another "frontier" in $T_{\mathtt{OPT}}$ and thus,

$$\sum_{u\in Y_1} p_u\cdot q_u + \sum_{u\in Y_2} p_u\cdot q_u \le \max_{u\in Y_1\cup Y_2} p_u \le p_r. \tag{14}$$

In addition, for every node $v\in X$, the set $Y_1\cap T_{L(v)}$ is a frontier in $T_{L(v)}$, and therefore $\sum_{u\in Y_1\cap T_{L(v)}} q_u \le p_v q_v$. This, combined with (13), implies

$$\sum_{u\in Y_1} q_u \le p_r. \tag{15}$$

By (2), we have

$$\begin{aligned}
\mathbf{E}_{\mathtt{OPT}}(G,t) &= \sum_{v\in T_{\mathtt{OPT}}} p_v\cdot q_v = \sum_{v\in X\cup Y_1\cup Y_2} p_v\cdot q_v + \sum_{v\in T_{\mathtt{OPT}}\setminus X\cup Y_1\cup Y_2} p_v\cdot q_v &(16)\\
&\le 2p_r + \sum_{u\in Y_1}\sum_{v\in T_u\setminus\{u\}} p_v\cdot q_v + \sum_{u\in X}\sum_{v\in T_{L(u)}\setminus T(Y_1)} p_v\cdot q_v\\
&\quad + \sum_{u\in Y_2}\sum_{v\in T_u\setminus\{u\}} p_v\cdot q_v + \sum_{u\in X}\sum_{v\in T_{R(u)}\setminus T(Y_2)} p_v\cdot q_v + \sum_{v\in T_{\mathtt{OPT}}\setminus T(X)} p_v\cdot q_v &(17)\\
&= 2p_r + \sum_{v\in Y_1} q_v\cdot\left(p_v\cdot\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)+(1-p_v)\cdot\mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right)\right)\\
&\quad + \sum_{u\in X}\sum_{v\in T_{L(u)}\setminus T(Y_1)} p_v\cdot q_v + \sum_{v\in Y_2} q_v\cdot\left(p_v\cdot\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)+(1-p_v)\cdot\mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right)\right)
\end{aligned}$$

$$+\sum_{u\in X}\sum_{v\in T_{R(u)}\setminus T(Y_2)} p_v\cdot q_v+\sum_{v\in T_{\mathtt{OPT}}\setminus T(X)} p_v\cdot q_v \tag{18}$$

$$\le 2p_r+\sum_{v\in Y_1} q_v\cdot\left(p_v\cdot \mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)+(1-p_v)\cdot\left(1+\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)\right)\right)$$

$$+\sum_{u\in X}\sum_{v\in T_{L(u)}\setminus T(Y_1)} p_v\cdot q_v+\sum_{v\in Y_2} q_v\cdot\left(p_v\cdot \mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right)+(1-p_v)\cdot \mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right)\right)$$

$$+\sum_{u\in X}\sum_{v\in T_{R(u)}\setminus T(Y_2)} p_v\cdot q_v+\sum_{v\in T_{\mathtt{OPT}}\setminus T(X)} p_v\cdot q_v \tag{19}$$

$$\le 3p_r+\sum_{v\in Y_1} q_v\cdot \mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)+\sum_{u\in X}\sum_{v\in T_{L(u)}\setminus T(Y_1)} p_v\cdot q_v$$

$$+\sum_{v\in Y_2} q_v\cdot \mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right)+\sum_{u\in X}\sum_{v\in T_{R(u)}\setminus T(Y_2)} p_v\cdot q_v+\sum_{v\in T_{\mathtt{OPT}}\setminus T(X)} p_v\cdot q_v. \tag{20}$$

In the above, (17) follows from (13) and (14) along with the fact that X, Y_1 and Y_2 are disjoint sets and by rearranging the second sum in (16). (18) follows from the definition of $T_{\mathtt{OPT}}$. (19) follows from Lemma 1. (20) follows from (15) and the fact that $1-p_v\le 1$ for all v.

D Example for the Edge-Weighted Case

An alternative direction where the stochastic matching problem discussed in this paper can be generalized is to assume that each edge in G has a weight, representing the *reward* we receive if the edge is part of the matching. The objective is to maximize the expected total reward. One might hope that a natural generalization of `GREEDY`, namely the algorithm that probes edges in the order of their probability times their weight, achieves a good approximation for this generalization. However, the following example shows that this is not the case.

Consider a graph G where its vertex set is composed of three disjoint components A, B, C, where $A=\{a_1,\dots,a_n\}$, $B=\{b_1,\dots,b_n\}$ and $C=\{c_1,\dots,c_n\}$. For each $i=1,\dots,n$, there is an edge (a_i,b_i) with weight and probability both being equal to 1. For each $i=1,\dots,n$ and $j=1,\dots,n$, there is an edge (b_i,c_j) with weight $\frac{n}{3\log n}$ and probability $\frac{2\log n}{n}$. The patience of every vertex is infinity. In this example, `GREEDY` will probe all edges between A and B and result in a solution of total weight n. However, `OPT` would try to probe all possible edges between B and C, where we know with at least a constant probability there is a perfect matching between B and C [10, 11]. In other words, the total expected weight of `OPT` is $\Omega(n^2/\log n)$.

E Adaptivity Gap

It can be seen that the greedy algorithm discussed in Section 3 is non-adaptive in the sense that it fixes an ordering of all pairs of nodes in the graph, and probes edges one by one according to this ordering (if the two endpoints are available). The optimal algorithm OPT, however, can be adaptive in general, i.e. the decision of the algorithm at each step can depend on the outcome of previous probes. Therefore, using the terminology of Dean et al. [8], Theorem 1 implies that the *adaptivity gap*, the ratio of the (expected) value of an optimal adaptive solution to the (expected) value of an optimal non-adaptive solution, is at most 4. On the other hand, the adaptivity gap is strictly larger than 1, as the following example shows.

Consider a graph $G = \{V, E\}$, where $V = \{v_1, \ldots, v_7, u\}$. Vertices $v_1, \ldots, v_7$ form a cycle of size 7, and the probability on the edges of the cycle is $p(v_1, v_2) = 0.2$ and everything else is 0.5. Further, u is connected to v_5 with $p(u, v_5) = 0.1$. Assume the patience of all vertices are infinite. It can be seen that the optimal adaptive strategy is to probe (u, v_5) first; upon a success, probe edges on the remaining path from one end to the other; and upon a fail, probe (v_1, v_2) secondly and follow the same strategy on the remaining path. The expected value of the algorithm is 2.21875. On the other hand, the optimal non-adaptive ordering, through a careful calculation, is $(u, v_5), (v_1, v_2), (v_2, v_3), \ldots, (v_7, v_1)$, which gives an expected value of 2.2175.

F Bad Example for Greedy Algorithm in Multiple Round Setting

Consider a bipartite graph $G = (A, B; E)$ where $A = \{\alpha_1, \ldots, \alpha_n\}$ and $B = \{\beta_1, \ldots, \beta_n\}$. Let $\epsilon = 1/n^3$. Let $p(\alpha_1, \beta_j) = \epsilon$ and $p(\alpha_i, \beta_1) = \epsilon$ for $i, j = 1, \ldots, n$, and $p(\alpha_i, \beta_i) = \frac{\epsilon}{n-2}$ for $i = 2, \ldots, n$. There are no other edges in the graph. Further, define patience $t(\alpha_1) = t(\beta_1) = \infty$ and $t(\alpha_i) = t(\beta_i) = 1$ for $i = 2, \ldots, n$. Consider any given $k \leq n - 1$. Now, a maximum matching in this example is $\{(\alpha_1, \beta_2), (\alpha_2, \beta_1), (\alpha_3, \beta_3), \ldots, (\alpha_n, \beta_n)\}$. The expected value that $\texttt{GREEDY}_k$ obtains by probing this matching in the first round is 3ϵ. After these probes, in the next round, due to patience restriction, in the best case only edge (α_1, β_1) will remain. Thus, $\texttt{GREEDY}_k$ obtains another ϵ, which implies that the total expected value is at most 4ϵ. On the other hand, consider another algorithm which probes edges (α_1, β_{i+1}) and (α_{i+1}, β_1) for any round $i = 1, \ldots, k$. The revenue generated by the algorithm is at least $2k\epsilon - 2\binom{n}{2}\epsilon^2 = \Omega(k\epsilon)$. (The second term $2\binom{n}{2}\epsilon^2$ is due to the following observation. For any pair of edges incident on α_1 or β_1, with probability ϵ^2 both of them are present. However, we can probe only one edge when they are incident to a same vertex.) Thus, $\texttt{GREEDY}_k$ can be as bad as a factor of $\Omega(k)$. Note that $\texttt{GREEDY}_k$ is trivially a factor k approximation algorithm.

G Proof of Theorem 2

We will prove the theorem by induction on k and the set of edges in the graph. For any subgraph $G' \subseteq G$, it is not hard to show that $\texttt{GREEDY}_k$ is in fact optimal for $k = 1$, and hence we have the induction basis. Assume the claim holds for any $1 \le \ell < k$, i.e. $\mathbf{E}_{\mathtt{OPT}_\ell}[G'] \le 4 \cdot \mathbf{E}_{\mathtt{GREEDY}_\ell}[G']$. Given the induction hypothesis, we will show $\mathbf{E}_{\mathtt{OPT}_k}[G] \le 4 \cdot \mathbf{E}_{\mathtt{GREEDY}_k}[G]$.

For any algorithm ALG, define a decision tree $T_{\mathtt{ALG}}$ as follows: Each node of $T_{\mathtt{ALG}}$ represents a tuple (G', E'), where G' is the current graph when ALG reaches that node and E' is the set of edges that ALG probes at that point. Each child of (G', E') corresponds to an outcome of different coin tosses of edges in E'. Thus, (G', E') has $2^{|E'|}$ children. Note that the height of $T_{\mathtt{ALG}}$ is at most k since we are only allowed to probe edges in k rounds.

We have the following two observations about $\mathbf{E}_{\mathtt{ALG}}$:

- For any edge $e \in E$, e contributes at most $p(e)$ to $\mathbf{E}_{\mathtt{ALG}}$. Consider all nodes in $T_{\mathtt{ALG}}$ that probe e, which forms a "frontier" in $T_{\mathtt{ALG}}$, the total contribution of these probes to $\mathbf{E}_{\mathtt{ALG}}$ is at most $p(e)$.
- For any vertex $v \in V$, v contributes at most one to $\mathbf{E}_{\mathtt{ALG}}$. That is, consider all nodes in $T_{\mathtt{ALG}}$ which probe edges incident to v, the total contribution of these probes to $\mathbf{E}_{\mathtt{ALG}}$ is at most one.

Assume the root of $T_{\mathtt{GREEDY}_k}$ represents (G, S), and its children represent $(G_1, E_1), \ldots, (G_\ell, E_\ell)$, respectively, where $\ell = 2^{|S|}$. Similarly, Assume the root of $T_{\mathtt{OPT}_k}$ represents (G, T), and its children represent $(H_1, F_1), \ldots, (H_m, F_m)$, respectively, where $m = 2^{|T|}$. Due to the greedy strategy, we know

$$\sum_{e \in S} p(e) \ge \sum_{e' \in T} p(e'). \tag{21}$$

For any G_i, $i = 1, \ldots, \ell$, let $\mathbf{Pr}[G_i]$ be the probability that $\mathtt{GREEDY}_k$ reaches G_i. In particular, assume $\mathtt{GREEDY}_k$ reaches (G_i, E_i) by succeeding on all edges in S_i and failing on all edges in S'_i, where $S'_i = S \setminus S_i$. Thus, $\mathbf{Pr}[G_i] = \prod_{e \in S_i} p(e) \cdot \prod_{e' \in S'_i}(1 - p(e'))$. Let $V(S_i)$ and $V(S'_i)$ be the set of vertices incident to edges in S_i and S'_i, respectively.

For any H_j, note that $(H_j \setminus S) \setminus V(S_i)$ is a subgraph[9] of G_i, which is a subgraph of G. Hence, by the induction hypothesis, we have

$$\mathbf{E}_{\mathtt{OPT}_{k-1}}((H_j \setminus S) \setminus V(S_i)) \le \mathbf{E}_{\mathtt{OPT}_{k-1}}(G_i) \le 4 \cdot \mathbf{E}_{\mathtt{GREEDY}_{k-1}}(G_i)$$

Therefore, according to the second observation above, the following is true for every $1 \le j \le n$ and $1 \le i \le \ell$:

$$\mathbf{E}_{\mathtt{OPT}_{k-1}}(H_j \setminus S) \le \mathbf{E}_{\mathtt{OPT}_{k-1}}((H_j \setminus S) \setminus V(S_i)) + 2|S_i| \le 4 \cdot \mathbf{E}_{\mathtt{GREEDY}_{k-1}}(G_i) + 2|S_i|. \tag{22}$$

Hence,

$$4 \cdot \mathbf{E}_{\mathtt{GREEDY}_k}(G) = 4 \cdot \sum_{e \in S} p(e) + 4 \cdot \sum_{i=1}^{\ell} \mathbf{Pr}[G_i] \cdot \mathbf{E}_{\mathtt{GREEDY}_{k-1}}(G_i)$$

[9] Here $(H_j \setminus S) \setminus V(S_i)$ denotes the graph obtained from H_j by first removing the edges in S and then by removing the vertices in $V(S_i)$ (and their incident edges).

$$
\begin{aligned}
&\geq \sum_{e' \in T} p(e') + 3 \cdot \sum_{e \in S} p(e) + 4 \cdot \sum_{i=1}^{\ell} \mathbf{Pr}[G_i] \cdot \mathbf{E}_{\mathtt{GREEDY_{k-1}}}(G_i) && (23)\\
&= \sum_{e' \in T} p(e') + \sum_{e \in S} p(e) + \sum_{i=1}^{\ell} \mathbf{Pr}[G_i] \cdot \big(4 \cdot \mathbf{E}_{\mathtt{GREEDY_{k-1}}}(G_i) + 2 \cdot |S_i|\big) \\
&\geq \sum_{e' \in T} p(e') + \sum_{e \in S} p(e) + \sum_{i=1}^{\ell} \mathbf{Pr}[G_i] \cdot \max_{j=1,\ldots,m} \mathbf{E}_{\mathtt{OPT_{k-1}}}(H_j \setminus S) && (24)\\
&\geq \sum_{e' \in T} p(e') + \sum_{e \in S} p(e) + \sum_{j=1}^{m} \mathbf{Pr}[H_j] \cdot \mathbf{E}_{\mathtt{OPT_{k-1}}}(H_j \setminus S) \\
&\geq \sum_{e' \in T} p(e') + \sum_{j=1}^{m} \mathbf{Pr}[H_j] \cdot \mathbf{E}_{\mathtt{OPT_{k-1}}}(H_j) && (25)\\
&= \mathbf{E}_{\mathtt{OPT_k}}(G),
\end{aligned}
$$

where $\mathbf{Pr}[H_j]$ is the probability that $\mathtt{OPT}_k$ reaches H_j. In the above (23) follows from (21). (24) follows from (22). Finally, (25) follows from the first observation above. This completes the proof of the theorem.

H Proof of Theorem 3

To prove the theorem, we will need the following lemma.

Lemma 3. *For any instance graph (G, t), assume $p(e) \leq p_{max} \leq 1$ for every edge e in G. For any vertex $v \in V$, define another instance (G, t') where $t'(v) = t(v) - 1$ and $t'(u) = t(u)$ for any $u \neq v$. Then $\mathbf{E}_{\mathtt{OPT_k}}[G] \leq \mathbf{E}_{\mathtt{OPT_k}}[G'] + p_{max}$.*

Proof. Consider $T_{\mathtt{OPT_k}}$ (recall the definition from the proof of Theorem 2) corresponding to G. For each root-leaf path in $T_{\mathtt{OPT_k}}$, consider the first node (if it exists) where there is a probe incident to v. Let S denote the set of all such nodes. Note that S defines a "frontier" on $T_{\mathtt{OPT_k}}$. Now consider a new algorithm $\mathtt{ALG}$ that works on $T_{\mathtt{OPT_k}}$ except that whenever $\mathtt{ALG}$ reaches to a node in S, it will only toss a coin for the edge incident to v (i.e. decide which children to choose next) but not probe that edge. It can be seen that $\mathbf{E}_{\mathtt{OPT_k}}[G] \leq \mathbf{E}_{\mathtt{ALG}}[G] + p_{max}$. Further, observe that when $\mathtt{ALG}$ runs on graph G, vertex v is probed at most $t(v) - 1$ times for any possible outcome of the probes. Thus, $\mathtt{ALG}$ is a valid algorithm for G' and $\mathbf{E}_{\mathtt{ALG}}[G] = \mathbf{E}_{\mathtt{ALG}}[G']$. The lemma follows by combining the two formulas and the fact that $\mathbf{E}_{\mathtt{ALG}}[G'] \leq \mathbf{E}_{\mathtt{OPT_k}}[G']$.

Proof of Theorem 3. The proof is similar to the proof of Theorem 2 and is by induction on the set of edges in the graph and patience parameters of the vertices. For any G' and G, we say G' is a *subgraph* of G if G' is an edge subgraph of G and for any vertex v, the patience of v in G' is smaller than or equal to that in G. For any subgraph $G' \subseteq G$, assume the claim holds for any $1 \leq \ell < k$, i.e. $\mathbf{E}_{\mathtt{OPT}_\ell}[G'] \leq$

$(4+2p_{max}/p_{min}) \cdot \mathbf{E}_{\mathtt{GREEDY}_\ell}[G']$. As before, the base case is trivial. Given the induction hypothesis, we will show $\mathbf{E}_{\mathtt{OPT}_\mathtt{k}}[G] \le (4+2p_{max}/p_{min}) \cdot \mathbf{E}_{\mathtt{GREEDY}_\mathtt{k}}[G]$.

We will be using notation from the proof of Theorem 2. For each G_i and H_j, let $H_j(i)$ be the graph obtained from H_j where the patience of each vertex in $V(S_i')$ is reduced by 1. By Lemma 3, we have

$$\mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[H_j \setminus S] \le \mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[H_j(i) \setminus S] + 2p_{max}|S_i'|. \tag{26}$$

For any $H_j(i)$, note that $(H_j(i) \setminus S) \setminus V(S_i)$ is a subgraph of G_i. Hence, by the induction hypothesis, we have

$$\mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[(H_j(i) \setminus S) \setminus V(S_i)] \le \mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[G_i] \le (4+2p_{max}/p_{min}) \cdot \mathbf{E}_{\mathtt{GREEDY}_{\mathtt{k}-1}}[G_i].$$

Therefore, using arguments similar to the ones we used to show (22), we have:

$$\mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[H_j(i) \setminus S] \le \mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[(H_j(i) \setminus S) \setminus V(S_i)] + 2|S_i| \le (4+2p_{max}/p_{min}) \cdot \mathbf{E}_{\mathtt{GREEDY}_{\mathtt{k}-1}}[G_i] + 2|S_i| \tag{27}$$

For each $i = 1, \ldots, \ell$, define

$$\pi(i) = \arg \max_{j=1,\ldots,m} \mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[H_j(i) \setminus S]$$

and

$$i^* = \arg \min_{i=1,\ldots,\ell} \mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[H_{\pi(i)}(i) \setminus S].$$

Now, $(4+2p_{max}/p_{min}) \cdot \mathbf{E}_{\mathtt{GREEDY}_\mathtt{k}}[G]$

$$= (4+2p_{max}/p_{min}) \cdot \sum_{e \in S} p(e) + (4+2p_{max}/p_{min}) \cdot \sum_{i=1}^{\ell} \mathbf{Pr}[G_i] \cdot \mathbf{E}_{\mathtt{GREEDY}_{\mathtt{k}-1}}[G_i]$$

$$\ge \sum_{e' \in T} p(e') + (3+2p_{max}/p_{min}) \cdot \sum_{e \in S} p(e) + (4+2p_{max}/p_{min}) \cdot \sum_{i=1}^{\ell} \mathbf{Pr}[G_i] \cdot \mathbf{E}_{\mathtt{GREEDY}_{\mathtt{k}-1}}[G_i] \tag{28}$$

$$= \sum_{e' \in T} p(e') + (1+2p_{max}/p_{min}) \cdot \sum_{e \in S} p(e) + \sum_{i=1}^{\ell} \mathbf{Pr}[G_i] \cdot \big((4+2p_{max}/p_{min}) \cdot \mathbf{E}_{\mathtt{GREEDY}_{\mathtt{k}-1}}[G_i] + 2 \cdot |S_i|\big)$$

$$\ge \sum_{e' \in T} p(e') + (1+2p_{max}/p_{min}) \cdot \sum_{e \in S} p(e) + \sum_{i=1}^{\ell} \mathbf{Pr}[G_i] \cdot \mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}} \left[H_{\pi(i)}(i) \setminus S\right] \tag{29}$$

$$\ge \sum_{e' \in T} p(e') + (1+2p_{max}/p_{min}) \cdot \sum_{e \in S} p(e) + \mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}} \left[H_{\pi(i^*)}(i^*) \setminus S\right] \tag{30}$$

$$\ge \sum_{e' \in T} p(e') + (1+2p_{max}/p_{min}) \cdot \sum_{e \in S} p(e) + \sum_{j=1}^{m} \mathbf{Pr}[H_j] \cdot \mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[H_j(i^*) \setminus S] \tag{31}$$

$$\ge \sum_{e' \in T} p(e') + \sum_{e \in S} p(e) + \sum_{j=1}^{m} \mathbf{Pr}[H_j] \cdot \big(\mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[H_j(i^*) \setminus S] + 2p_{max}|S_{i^*}'|\big) \tag{32}$$

$$\ge \sum_{e' \in T} p(e') + \sum_{e \in S} p(e) + \sum_{j=1}^{m} \mathbf{Pr}[H_j] \cdot \mathbf{E}_{\mathtt{OPT}_{\mathtt{k}-1}}[H_j \setminus S] \tag{33}$$

$$\geq \sum_{e' \in T} p(e') + \sum_{j=1}^{m} \mathbf{Pr}[H_j] \cdot \mathbf{E}_{\texttt{OPT}_{k-1}}[H_j] \tag{34}$$
$$= \mathbf{E}_{\texttt{OPT}_k}[G].$$

In the above, (28) follows from the fact that $\texttt{GREEDY}_k$ probes the maximum matching in every round. (29) follows from (27) and the definition of $\pi(i)$. (30) follows from the definition of i^*. (31) follows from the definition of $\pi(i^*)$. (32) follows from the fact that $2p_{max}/p_{min} \sum_{e \in S} p(e) \geq 2p_{max}|S| \geq 2p_{max}|S'_{i^*}|$. (33) follows from (26). Finally, (34) follows from the first observation in the proof of Theorem 2. □

I Missing Details from Section 4, Further Extension

I.1 The General Case

We will now see how Theorem 3 can be used to obtain a (randomized) approximation algorithm for the general case of the multi-round stochastic matching problem with an approximation factor of $O(\log n)$. Given an instance (G, t), denote the maximum probability of an edge in this instance by $p_{\max}$. First, we note that $\mathbf{E}_{\texttt{OPT}}(G, t) \geq p_{\max}$, and therefore removing all edges that have probability less than $p_{\max}/n^2$ cannot decrease the value of $\mathbf{E}_{\texttt{OPT}}(G, t)$ by more than a constant factor. Next, we partition the set of edges of G into $O(\log n)$ classes, depending on which of the intervals $(p_{\max}/2, p_{\max}]$, $(p_{\max}/4, p_{\max}/2]$, $\ldots$, $(p_{\max}/2^{2\lceil \log_2 n \rceil}, p_{\max}/2^{2\lceil \log_2 n \rceil} - 1]$, the probability of the edge falls into. Let E_i denote the i'th class of this partition. The algorithm then picks one of the edge sets E_i at random and then runs $\texttt{GREEDY}_k$ on the instance obtained by restricting the edges of G to E_i. Note that we can write the value of $\texttt{OPT}$ as the sum of $O(\log n)$ terms, where the i'th term corresponds to the value that $\texttt{OPT}$ extracts from edges in E_i (call this value $\texttt{OPT}_i$). Further, by Theorem 3, $\texttt{GREEDY}_k$ obtains a value at least $\texttt{OPT}_i/8$. The claimed $O(\log n)$ approximation factor follows by recalling that each of the $O(\log n)$ choices of E_i was picked uniformly at random.

I.2 Further Extension

The extension of $\texttt{GREEDY}_k$ in this model is straightforward: in each round probe the maximum weighted matching (with the constraint that the matching uses at most C edges).

Theorems 2 and 3 also hold in the new model because the only place we use the property of the greedy algorithm in the analysis is the claim that the expected number of matched edges in the first round of the greedy algorithm is at least the expected number of matched edges in any round of the optimal algorithm. This property still holds as long as both the greedy and the optimal algorithm probe at most C edges in any round.

The bad example in Appendix F for $\texttt{GREEDY}_k$ can be extended to show that the greedy algorithm in the new model has an approximation factor of

$\Omega(\min(k, C))$. To see this, consider the previous bad example with $k = n - 1$. Further, $p(\alpha_i, \beta_i) = \frac{\epsilon}{C-2}$ for $1 < i \le n$. Everything else in the graph remains the same. Again it can be shown that the optimal gets a value of at least $\Omega(k\epsilon)$. On the other hand, the greedy algorithm in any round can match at most 3ϵ many edges in expectation. Further, the greedy algorithm will be left with an empty graph after $\lceil n/C \rceil = \lceil (k+1)/C \rceil$ rounds. Thus, greedy in total can obtain a revenue of at most $3\epsilon \cdot \min(\lceil (k+1)/C \rceil, k)$, which is $O(\epsilon)$ if $C > k$ and $O(k\epsilon/C)$ otherwise. This gives an approximation factor of $\Omega(\min(k, C))$ as claimed.

Of course, the greedy algorithm is still a k-approximation algorithm in the new model. We will now briefly argue that the proof of Theorem 3 can be extended to show that the greedy algorithm is also a $(2C+4)$-approximation algorithm. Lemma 3 can be easily extended to show that $\mathbf{E}_{\mathtt{OPT_k}}[G] \le \mathbf{E}_{\mathtt{OPT_k}}[G'] + p_{max}$, where p_{max} is the maximum probability on any edge in G. This in turn implies that (26) can be replaced by

$$\mathbf{E}_{\mathtt{OPT_{k-1}}}[H_j \setminus S] \le \mathbf{E}_{\mathtt{OPT_{k-1}}}[H_j(i) \setminus S] + 2p_{max}|S'_i|.$$

Further, by the property of the greedy algorithm, we have

$$\sum_{e \in S} p(e) \ge p_{max}.$$

Finally, using the above two facts along with the fact that $|S'_i| \le |S| \le C$ one can easily extend the arguments in proof of Theorem 3 to show that $(2C + 4)\mathbf{E}_{\mathtt{GREEDY}k}[G] \ge \mathbf{E}_{\mathtt{OPT}}[G]$.

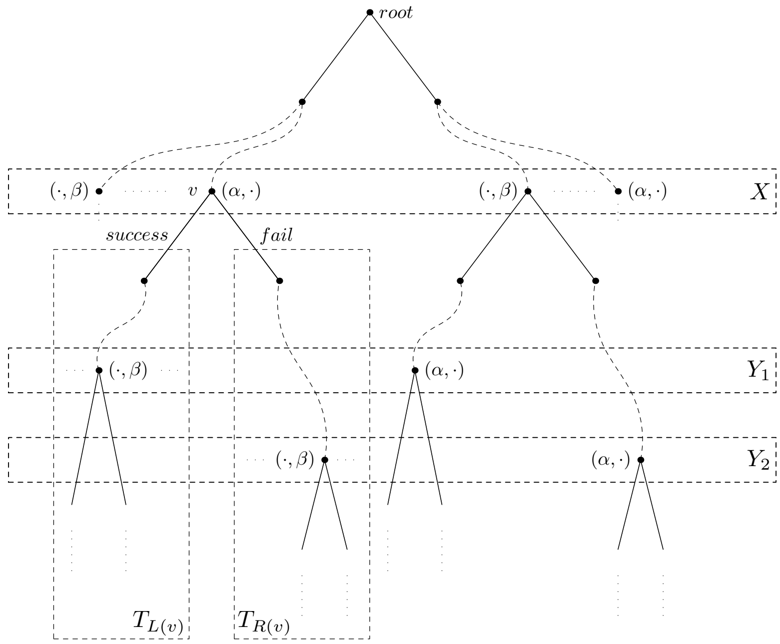

Fig. 3. Definition of X, Y_1, Y_2 in $T_{\texttt{OPT}}$.