Approximating Matches Made in Heaven
(Extended Abstract)

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, 14–16]. 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. In addition, 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 [10, 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. Actually, 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. 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.[1] 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 [12, 13], 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 [15, 16]. 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*), the transplant can not be performed. Otherwise (the crossmatch is *negative*), the transplant may be performed.[2]

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

[1] 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`.

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

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 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 $p(\alpha, \beta) \in [0, 1]$.[3] For the rest of the paper, without loss of generality we 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.

[3] 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.

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

Decision Tree Representation. For any deterministic algorithm $\mathtt{ALG}$ and any instance (G, t) of the problem, the entire operation of $\mathtt{ALG}$ on (G, t) can be represented as an (exponential-sized) *decision tree* $T_{\mathtt{ALG}}$. The root of $T_{\mathtt{ALG}}$, r, represents the first pair $e = (\alpha, \beta) \in E$ probed by $\mathtt{ALG}$. 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 $v \in T_{\mathtt{ALG}}$, 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 $v \in T_{\mathtt{ALG}}$, 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 $\mathtt{ALG}'$ on the sub-instance (G_v, t_v) corresponding to v. Define $\mathbf{E}_{\mathtt{ALG}}(T_v)$ to be the expected value generated by the algorithm corresponding to $\mathtt{ALG}'$, i.e. $\mathbf{E}_{\mathtt{ALG}}(T_v) = \mathbf{E}_{\mathtt{ALG}'}(G_v, t_v)$.

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 [11]). 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 $\mathtt{OPT}$ the optimal algorithm, i.e., the solution of this MDP. Note that we can assume without loss of generality that $\mathtt{OPT}$ is deterministic, and therefore, a decision tree $T_{\mathtt{OPT}}$ representing $\mathtt{OPT}$ 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* $v \in T_{\mathtt{OPT}}$, even for nodes v that have probability zero of being reached. Note that such nodes can exist in $T_{\mathtt{OPT}}$, since $\mathtt{OPT}$ can probe edges of probability 1, in which case the corresponding right subtree is never reached.

It is not even known if 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, in the full version of the paper, we show that the gap between OPT and the optimal offline solution can be unbounded.

3 Greedy Algorithm

We consider the following greedy algorithm.

```
GREEDY
 1. Sort all edges in E by probabilities, say, p(e_1) ≥ p(e_2) ≥ ··· ≥
    p(e_m) (ties are broken arbitrarily)
 2. For i = 1,...,m, 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 based on carefully charging the value obtained at different nodes of $T_{\mathtt{OPT}}$ to $T_{\mathtt{GREEDY}}$. We begin by stating two lemmas that will be useful for the proof.

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. (The proofs of the lemmas are deferred to the full version of the paper.)

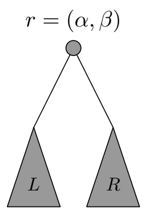

Fig. 1. Greedy tree $T_{\mathtt{GREEDY}}$

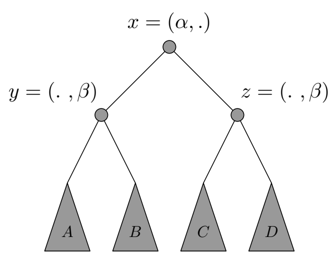

Fig. 2. Optimum tree $T_{\mathtt{OPT}}$

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 $T_{\texttt{GREEDY}}$ as in Figure 1 and suppose that $T_{\texttt{OPT}}$ is as in Figure 2. Let p_r be the probability of success of probe (α, β).

Note that the algorithm $\texttt{ALG}_1$ defined by subtree A in $T_{\texttt{OPT}}$ is a valid algorithm for the left subtree L 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 L of $T_{\texttt{GREEDY}}$). Furthermore, $\texttt{ALG}_1$ achieves the same value, in expectation, as the optimum algorithm on subtree A. Similarly the algorithm $\texttt{ALG}_2$ defined by subtree D in $T_{\texttt{OPT}}$ is a valid algorithm for the right subtree R of greedy *except* $\texttt{ALG}_2$ may perform a probe to (α, β). Thus we define a secondary (randomized) algorithm $\texttt{ALG}'_2$ which follows $\texttt{ALG}_2$ 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 $\texttt{ALG}'_2$ is a valid algorithm for the right subtree R 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 $\texttt{ALG}_1$ and $\texttt{ALG}'_2$ on the left and right subtree L and R 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 that 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 that $\mathbf{E}_{\texttt{OPT}}(G, t) \leq 4 \cdot \mathbf{E}_{\texttt{GREEDY}}(G, t)$.

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 sub-instances corresponding to the left and right child of r, respectively. Note that

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

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 sub-instance of (G, t) obtained by removing edge (α, β). Then,

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

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 $0 < p_v \leq p_r < 1$ by the definition of the greedy algorithm.

We define q_v to be the probability that 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:

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

Define $X \subseteq T_{\texttt{OPT}}$ to be the set of nodes that correspond to the first time where 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 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. 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 $R(v)$.

For any subset of nodes $S \subseteq T_{\texttt{OPT}}$, define $T(S) = \bigcup_{v \in S} T_v$. We show in the full version of the paper that

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

Define an algorithm $\mathtt{ALG}_1$ that works as follows: $\mathtt{ALG}_1$ follows the decision tree of $\mathtt{OPT}$ 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 $\mathtt{ALG}_1$, every path from the root to a node in $T(Y_1)$ (and $\bigcup_{u\in X} T_{L(u)} \setminus T(Y_1)$) has two (and one respectively) less successful probes in $X \cup Y_1$ than $\mathtt{OPT}$, it follows that

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

(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

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

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

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

(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). That is, the contribution of edge (α, β) is not included in $\mathtt{ALG}'_2$. It is easy to see that

$$\mathbf{E}_{\mathtt{ALG}_2} \le \mathbf{E}_{\mathtt{ALG}'_2} + p_r \tag{7}$$

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

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

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 $\mathbf{E}_{\mathtt{OPT}}(G,t)$ but it clearly does not change the value of $\mathtt{ALG}_1$ or $\mathtt{ALG}_2$. Let $\mathbf{E}_{\mathtt{OPT}}(G,t)'$ be the value of the algorithm $T_{\mathtt{OPT}}$ on this new instance. From (3), we have

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

where the last line follows from (4) and (6). Therefore, we have

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

where (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. In the full version of the paper, we show 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
    (a) compute the maximum weighted matching in the current graph
    (b) probe all edges in the matching
```

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

However, we can still prove that $\mathtt{GREEDY}_k$ 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 can show that $\texttt{GREEDY}_k$ is a 4-approximation.

Theorem 2 *For any graph* $G = (V, E)$*,* $\mathbf{E}_{\texttt{OPT}_k}(G) \le 4 \cdot \mathbf{E}_{\texttt{GREEDY}_k}(G)$*, 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.

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* $0 < p_{\min} \le p_{\max} \le 1$*. Then* $\mathbf{E}_{\texttt{OPT}_k}(G) \le (4 + 2p_{\max}/p_{\min}) \cdot \mathbf{E}_{\texttt{GREEDY}_k}(G)$*.*

Note that this implies that $\texttt{GREEDY}_k$ is a 6-approximation in the uniform probability case, i.e. $p_{\min} = p_{\max}$.

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 $i \ge 0$.

A Further Extension. We also consider the following extension of the multi-round model. In each round, an algorithm is only allowed to probe a matching of size at most C, where $1 \le C \le \lfloor |V|/2 \rfloor$ 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, for the arbitrary patience and probability case, $\texttt{GREEDY}_k$ is a $\Theta(\min(k, C))$-approximation algorithm. (The details are deferred to the full version of the paper.)

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). Actually, we can 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 of 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 the full version of the paper, we exhibit an example that shows that the natural greedy algorithm has an unbounded approximation ratio. In addition, 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 what is the "gap" between the non-adaptive and *adaptive* optimal values (e.g. [8])? In the full version of the paper, we present an example to 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, 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, 2005.
5. A. Abdulkadiroglu and T. Sonmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 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)*, pages 552–563, 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. C.H. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150, 1991.
11. M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley Series in Probability and Statistics, 2005.
12. F.T. Rapaport. The case for a living emotionally related international kidney donor exchange registry. *Transplantation Proceedings*, 18:5–9, 1986.
13. 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*, 336(24):1752–1755, 1997.
14. 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.
15. A.E. Roth, T. Sonmez, and U. Unver. Kidney exchange. *Quarterly Journal of Economics*, 119(2):457–488, 2004.
16. A.E. Roth, T. Sonmez, and U. Unver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151–188, 2005.