Noname manuscript No.
(will be inserted by the editor)

When LP is the Cure for Your Matching Woes: Improved Bounds for Stochastic Matchings

**Nikhil Bansal · Anupam Gupta · Jian Li · Julián Mestre ·
Viswanath Nagarajan · Atri Rudra**

Abstract Consider a random graph model where each possible edge e is present independently with some probability p_e. Given these probabilities, we want to build a large/heavy matching in the randomly generated graph. However, the only way we can find out whether an edge is present or not is to query it, and if the edge is indeed present in the graph, we are forced to add it to our matching. Further, each vertex i is allowed to be queried at most t_i times. How should we *adaptively* query the edges to maximize the expected weight of the matching? We consider several matching problems in this general framework (some of which arise in kidney exchanges and online dating, and others arise in modeling online advertisements); we give LP-rounding based constant-factor approximation algorithms for these problems. Our main results are the following:

- We give a 4 approximation for weighted stochastic matching on general graphs, and a 3 approximation on bipartite graphs. This answers an open question from [Chen *et al.* ICALP 09].
- We introduce a generalization of the stochastic *online* matching problem [Feldman *et al.* FOCS 09] that also models preference-uncertainty and timeouts of buyers, and give a constant factor approximation algorithm.

1 Introduction

Motivated by applications in kidney exchanges and online dating, Chen *et al.* [9] proposed the following stochastic matching problem: we want to find a maximum matching in a random graph G on n nodes, where each edge $(i, j) \in [\binom{n}{2}]$ exists with probability p_{ij}, independently of the other edges. However, all we are given are the probability values $\{p_{ij}\}$. To find out whether the random graph G has the edge (i, j) or not, we have to try to add the edge (i, j) to our current matching (assuming that i and j are both unmatched in our current partial matching)—we call this "probing" edge (i, j). As a result of the probe, we also find out if (i, j) exists or not—and if the edge (i, j) indeed exists in the random graph G, it gets irrevocably added to M. Such policies make sense, e.g., for dating agencies, where the only way to find out if two people are actually compatible is to send them on a date; moreover, if they do turn out to be compatible, then it makes sense to match them to each other. Finally, to model the fact that there might be a limit on the number of unsuccessful dates a person might be willing to participate in, "timeouts" on vertices are also provided. More precisely, valid policies are allowed, for each vertex i, to only probe at most t_i edges incident to i. Similar considerations arise in kidney exchanges, details of which appear in [9].

Nikhil Bansal · Viswanath Nagarajan
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA.

Anupam Gupta
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Supported in part by NSF awards CCF-0729022 and CCF-1016799, and an Alfred P. Sloan Fellowship.

Jian Li
Computer Science Department. University of Maryland College Park, MD, USA.

Julián Mestre
School of Information Technologies, University of Sydney, NSW, Australia.

Atri Rudra
Department of Computer Science and Engineering, University at Buffalo, SUNY, Buffalo NY 14260, USA. Supported by NSF CAREER award CCF-0844796.

Chen *et al.* asked the question: how can we devise probing policies to maximize the expected cardinality (or weight) of the matching? They showed that the greedy algorithm that probes edges in decreasing order of p_{ij} (as long as their endpoints had not timed out) was a 4-approximation to the cardinality version of the stochastic matching problem. Quite recently, Adamczyk has proved that the greedy algorithm is a 2-approximation for unweighted stochastic matching [1]. However, this greedy algorithm (and other simple greedy schemes) can be seen to be arbitrarily bad in the presence of weights, and they left open the question of obtaining good algorithms to *maximize the expected weight* of the matching produced. In addition to being a natural generalization, weights can be used as a proxy for revenue generated in matchmaking services. (The unweighted case can be thought of as maximizing the social welfare.) In this paper, we resolve the main open question from Chen *et al.* [9] by obtaining constant approximations for the weighted stochastic matching problems.

First, we consider a more general problem, called *stochastic k-set-packing*, where we try to pack k-hyperedges with random sizes and profits into a d-dimensional knapsack of a given size. The stochastic k-set-packing problem is a direct generalization of the stochastic matching problem (for $k = 4$; See the reduction in Section 2). We also note that this is a slight generalization of the stochastic b-matching problem of [10]. In particular, our model allows *correlations* between the profit of an item and its size-vector, whereas in [10] the profit of each item is fixed (or independent of its size-vector). Indeed, it is the discreteness in the sizes (i.e. $0-1$ values) that allows the LP-based approach to work for stochastic k-set-packing; if the instantiations were allowed to be in $[0, 1]$ then the LP has a large integrality gap even with just one constraint (see eg. in Appendix A of [18]). Moreover, our focus is on the situation where $k \ll d$. For this setting of parameters, we improve on the $\sqrt{d}$-approximation of [10] (which only holds for independent profits and sizes) by showing the following (Section 2).

Theorem 1 *There is a $2k$-approximation algorithm for the weighted stochastic k-set-packing problem. When the column outcomes are monotone, there is a $k + 1$ approximation algorithm.*

Our main idea is to use the knowledge of item probabilities to solve a linear program where each item e has a variable $0 \leq y_e \leq 1$ corresponding to the probability that a strategy packs e (over all possible realizations of the hypergraph). This is similar to the approach for stochastic packing problems considered by Dean et al. [11,10]. Our improved approximation for monotone column outcomes is obtained using the FKG inequality to strengthen the probability bound. Our usage of the FKG inequality is similar to that in [29]. The second part of Theorem 1 also implies a simple 5-approximation for stochastic matching. However, using more structure in the matching problem, we could obtain the following better approximation ratios.

Theorem 2 *There is a 4-approximation algorithm for the weighted stochastic matching problem. For bipartite graphs, there is a 3-approximation algorithm.*

The improved approximations use the same linear program as before, but more involved rounding methods to decide which edges to probe. The rounding procedure for bipartite graphs uses *dependent rounding* [14] on the y-values to obtain a set $\widehat{E}$ of edges to be probed, and then probes edges of $\widehat{E}$ in a uniformly random order. For non-bipartite graphs, the algorithm first samples a random bipartite subgraph and then applies the bipartite rounding algorithm on it.

The probing strategy returned by the algorithm can in fact be made *matching-probing* [9]. In this alternative (more restrictive) probing model we are given an additional parameter k and edges need to be probed in k rounds, each round being a matching. It is clear that this matching-probing model is more restrictive than the usual *edge-probing* model (with timeouts $\min\{t_i, k\}$) where one edge is probed at a time. Our algorithm obtains a matching-probing strategy that is only a small constant factor worse than the optimal edge-probing strategy; hence, we also obtain the same constant approximation guarantee for weighted stochastic matching in the matching-probing model. It is worth noting that previously only a logarithmic approximation in the unweighted case was known [9].

Theorem 3 *There is a 4-approximation algorithm for the weighted stochastic matching problem in the* matching-probing *model. For bipartite graphs, there is a 3-approximation algorithm.*

Apart from solving these open problems and yielding improved approximations, our LP-based analysis turns out to be applicable in a wider context.

Online *Stochastic Matching with Timeouts.* In a bipartite graph $(A, B; E)$ of items $i \in A$ and potential buyer types $j \in B$, p_{ij} denotes the probability that a buyer of type j will buy item i. A sequence of n buyers are to arrive online, where the type of each buyer is an i.i.d. sample from B according to some pre-specified distribution—when a buyer of type j appears, he can be shown a list L of up to t_j as-yet-unsold items, and the buyer buys the *first* item on the list according to the given probabilities $p_{\cdot,j}$. (Note that with probability $\prod_{i \in L}(1 - p_{ij})$, the buyer leaves without buying anything.) What items should we show buyers when they arrive online, and in which order, to maximize the expected weight of the matching? Building on the algorithm for stochastic matching in Section 2, we prove the following in Section 4.

Theorem 4 *There is a 7.92-approximation algorithm for the online stochastic matching problem with timeouts.*

This question is an extension of similar online stochastic matching questions considered earlier in [12]—in that paper, $w_{ij}, p_{ij} \in \{0, 1\}$ and $t_j = 1$. Our model tries to capture the facts that buyers may have a limited attention span (using the timeouts), they might have uncertainties in their preferences (using edge probabilities), and that they might buy the first item they like rather than scanning the entire list.

A New Proof for Greedy. The proof in [9] that the greedy algorithm for stochastic matching was a 4-approximation in the unweighted case was based on a somewhat delicate charging scheme involving the decision trees of the algorithm and the optimal solution. We show (Appendix B) that the greedy algorithm, which was defined without reference to any LPs, admits a simple LP-based analysis.

Theorem 5 *The greedy algorithm is a 5-approximation for the unweighted stochastic matching problem.*

Cardinality Constrained Matching in Rounds. We also consider the model from [9] where one can probe as many as C edges in parallel, as long as these C edges form a matching; the goal is to maximize the expected weight of the matched edges after k rounds of such probes. We improve on the $\min\{k, C\}$-approximation offered in [9] (which only works for the unweighted version), and show in Appendix A:

Theorem 6 *There is a constant-factor approximation algorithm for weighted cardinality constrained multiple-round stochastic matching.*

1.1 Related Work

Perhaps the work most directly related to this work are those on stochastic knapsack problems (Dean et al. [11, 10]) and multi-armed bandits (see [16,17] and references therein). Also related is some recent work [6] on budget constrained auctions, which uses similar LP rounding ideas.

In recent years stochastic optimization problems have drawn much attention from the theoretical computer science community where stochastic versions of several classical combinatorial optimization problems have been studied. Some general techniques have also been developed [19,30]. See [31] for a survey.

The online bipartite matching problem was first studied in the seminal paper by Karp *et al.* [23] and an optimal $1 - 1/e$ competitive online algorithm was obtained. Katriel *et al.* [24] considered the two-stage stochastic min-cost matching problem. In their model, we are given in a first stage probabilistic information about the graph and the cost of the edges is low; in a second stage, the actual graph is revealed but the costs are higher. The original online stochastic matching problem was studied recently by Feldman *et al.* [12]. They gave a 0.67-competitive algorithm, beating the optimal $1 - 1/e$-competitiveness known for worst-case models [23,22,27,7,15]. Recently, some improved bounds on this model were obtained [3,26]. Our model differs from that in having a bound on the number of items each incoming buyer sees, that each edge is only present with some probability, and that the buyer scans the list linearly (until she times out) and buys the first item she likes.

Our problem is also related to the Adwords problem [27], which has applications to sponsored search auctions. The problem can be modeled as a bipartite matching problem as follows. We want to assign every vertex (a query word) on one side to a vertex (a bidder) on the other side. Each edge has a weight, and there is a budget on each bidder representing the upper bound on the total weight of edges that may be assigned to it. The objective is to maximize the total revenue. The stochastic version in which query words arrive according to some known probability distribution has also been studied [25].

For the k-set packing problem, it is known that the simply greedy algorithm provides a k-approximation and an improvement in the ratio, to $\frac{k}{2}$ can be obtained by a local search heuristic [21], which is also the best known approximation to date. Recently, $O(k)$-approximations were obtained for the more general k-column sparse packing problem (the entries of the matrix can be arbitrary positive numbers rather than just $0/1$) [5]. It is also known that the k-set packing problem can not be efficiently approximated to within a factor of $\Omega(\frac{k}{\ln k})$ unless $\mathsf{P} = \mathsf{NP}$ [20]. This is also a lower bound for our stochastic k-set packing problem. Additionally for LP-based approaches (as in this paper) k-set packing has an integrality gap of $k - 1 + \frac{1}{k}$ [13].

1.2 Preliminaries

For any integer $m \geq 1$, define $[m]$ to be the set $\{1, \ldots, m\}$. For a maximization problem, an α-approximation algorithm is one that computes a solution with expected objective value at least $1/\alpha$ times the expected value of the optimal solution.

We must clarify here the notion of an optimal solution. In standard worst case analysis we would compare our solution against the optimal *offline* solution, e.g. the value of the maximum matching, where the offline knows all the edge instantiations in advance (i.e. which edge will appear when probed, and which will not). However, it can be easily verified that due to the presence of timeouts, this adversary is too strong [9]. Consider the following example. Suppose we have a star where each vertex has timeout 1, and each edge has $p_{ij} = 1/n$. The offline optimum can match an edge whenever the star has an edge i.e. with probability about $1 - 1/e$, while our algorithm can only get expected $1/n$ profit, as it can only probe a single edge. Hence, for all problems in this paper we consider the setting where even the optimum does not know the exact instantiation of an edge until it is probed. This gives our algorithms a level playing field. The optimum thus corresponds to a "strategy" of probing the edges, which can be chosen from an exponentially large space of potentially adaptive strategies.

We note that our algorithms in fact yield *non-adaptive* strategies for the corresponding problems, that are only constant factor worse than the adaptive optimum. This is similar to previous results on stochastic packing problems: knapsack (Dean *et al.* [11,10]) and multi-armed bandits (Guha-Munagala [16,17] and references therein).

2 Stochastic k-Set Packing

We first consider a generalization of the stochastic matching problem to hypergraphs, where each edge has size at most k. Formally, the input to this *stochastic k-set packing* problem consists of

- n items/columns, where each item has a random profit $v_i \in \mathbb{R}_+$, and a random d-dimensional size $S_i \in \{0, 1\}^d$; these random values and sizes are drawn from a probability distribution specified as part of the input. We note that the size-vector S_i and profit v_i of each item i are allowed to be correlated (this is what distinguishes our model from [10]). The probability distributions for different items are independent. Additionally, for each item, there is a set C_i of at most k coordinates such that each size vector takes positive values only in these coordinates; i.e., $S_i \subseteq C_i$ with probability 1 for each item i.
- A capacity vector $b \in \mathbb{Z}_+^d$ into which the items must be packed.

The parameter k is called the *column sparsity* of the problem. The instantiation of any column (i.e., its size and profit) is known only when it is probed. The goal is to compute an adaptive strategy of choosing items until there is no more available capacity such that the expectation of the obtained profit is maximized.

Note that the stochastic matching problem can be modeled as a stochastic 4-set packing problem in the following way: we set $d = 2n$, and associate the i^{th} and $(n+i)^{th}$ coordinate with the vertex i—the first n coordinates capture whether the vertex is free or not, and the second n coordinates capture how many probes have been made involving that vertex. For any $t \in [d]$, let $e_t \in \{0, 1\}^d$ denote the indicator vector with a single 1 in the t^{th} position. Now each edge (i, j) is an item which has the following distribution: with probability p_{ij} the value is w_{ij} and size is $e_i + e_j + e_{n+i} + e_{n+j}$, and with remaining probability $1 - p_{ij}$ the value is 0 and size is $e_{n+i} + e_{n+j}$. Note that for each item, its size and value are correlated. If we set the capacity vector to be $b = (1, 1, \cdots, 1, t_1, t_2, \cdots, t_n)$, this precisely captures the stochastic matching problem. In this special case each size vector has $\leq k = 4$ ones.

This stochastic k-set packing problem was studied (among many others) as the "stochastic b-matching" problem in Dean *et al.* [10]; however their model assumed deterministic values of items, so their results do not apply here directly. Moreover the authors of that work did not consider the 'column sparsity' parameter k and instead gave an $O(\sqrt{d})$-approximation algorithm for the general case. Here we consider the performance of algorithms for this problem specifically as a function of the column sparsity k, and prove Theorem 1.

A quick aside about "safe" and "unsafe" adaptive policies: a policy is called *safe* if it can include an item only if there is *zero* probability of violating any capacity constraint. In contrast, an *unsafe* policy may attempt to include an item even if there is non-zero probability of violating capacity—however, if the random size of the item causes the capacity to be violated, then no profit is received for the overflowing item, and moreover, no further items may be included by the policy. The model in Dean et al. [10] allowed unsafe policies, whereas we are interested in safe policies. However, due to the discreteness of sizes in stochastic k-set packing, it can be shown that our approximation guarantee is relative to the optimal unsafe policy (see Subsection 2.2).

For each item $i \in [n]$ and constraint $j \in [d]$, let $\mu_i(j) := \mathbb{E}[S_i(j)]$, the expected value of the j^{th} coordinate in size-vector S_i. For each column $i \in [n]$, the coordinates $\{j \in [d] \mid \mu_i(j) > 0\}$ are called the *support* of column i.

By column sparsity, the support of each column has size at most k. Also, let $w_i := \mathbb{E}[v_i]$, the mean profit, for each $i \in [n]$. We now consider the natural LP relaxation for this problem, as in [10].

$$\text{maximize} \sum_{i=1}^{n} w_i \cdot y_i \tag{LP1}$$

subject to

$$\sum_{i=1}^{n} \mu_i(j) \cdot y_i \le b_j \qquad \forall j \in [d] \tag{1}$$

$$y_i \in [0,1] \qquad \forall i \in [n] \tag{2}$$

The following claim shows that the LP above is a valid relaxation for the stochastic k-set-packing problem.

Claim 1 *The optimal value for* (LP1) *is an upper bound on any (adaptive) algorithm for stochastic k-set-packing.*

Proof: Let p_i be the probability that an adaptive strategy A packs item i. To show the claim, it suffices to show that p_is satisfy the constraints (1) for any adaptive strategy A. Consider the jth constraint. Conditioned on any instantiation of all items, A can pack at most b_j items for which $\mu_i(j) = 1$, since A is a safe policy. Hence these constraints hold unconditionally as well, which implies that any valid strategy satisfies (1). ∎

Let y^* denote an optimal solution to this linear program, which in turn gives us an upper bound on any adaptive (safe) strategy. Our rounding algorithm proceeds as follows. Fix a constant $\alpha \ge 1$, to be specified later. The algorithm picks a uniformly random permutation $\pi : [n] \to [n]$ on all columns, and probes only a subset of the columns as follows. At any point in the algorithm, column c is *safe* iff there is positive residual capacity in *all* the coordinates in the support of c—in other words, irrespective of the instantiation of S_c, it can be feasibly packed with the previously chosen columns. The algorithm inspects columns in the order of π, and whenever it is safe to probe the next column $c \in [n]$, it does so with probability $\frac{y_c}{\alpha}$. Note that the algorithm skips all columns that are unsafe at the time they appear in π.

We now prove the first part of Theorem 1 by showing that this algorithm is a $2k$-approximation for a suitable value of α. For any column $c \in [n]$, let $\{\mathbf{I}_{c,\ell}\}_{\ell=1}^{k}$ denote the indicator random variables for the event that the ℓ^{th} constraint in the support of c is tight at the time when c is considered under the random permutation π. Note that the event "column c is safe when considered" is precisely $\bigwedge_{\ell=1}^{k} \neg \mathbf{I}_{c,\ell}$. By a trivial union bound, the $\Pr[\, c \text{ is safe}\,] \ge 1 - \sum_{\ell=1}^{k} \Pr[\mathbf{I}_{c,\ell}]$.

Lemma 1 *For any column $c \in [n]$ and index $\ell \in [k]$,* $\Pr[\mathbf{I}_{c,\ell}] \le \frac{1}{2\alpha}$.

Proof: Let $j \in [d]$ be the ℓ^{th} constraint in the support of c. Let U_c^j denote the usage of constraint j, when column c is considered (according to π). We have:

$$\begin{aligned}
\mathbb{E}[U_c^j] &= \sum_{a=1}^{n} \Pr[\,\text{column } a \text{ appears before } c \text{ AND } a \text{ is probed}\,] \cdot \mu_a(j), \\
&\le \sum_{a=1}^{n} \Pr[\,\text{column } a \text{ appears before } c\,] \cdot \frac{y_a}{\alpha} \cdot \mu_a(j), \\
&= \sum_{a=1}^{n} \frac{y_a}{2\alpha} \cdot \mu_a(j), \\
&\le \frac{b_j}{2\alpha}.
\end{aligned}$$

Since $\mathbf{I}_{c,\ell} = \{U_c^j \ge b_j\}$, Markov's inequality implies that $Pr[\mathbf{I}_{c,\ell}] \le \mathbb{E}[U_c^j]/b_j \le \frac{1}{2\alpha}$. ∎

Again using the trivial union bound, the probability that a particular column c is safe when considered under π is at least $1 - \frac{k}{2\alpha}$, and thus the probability of actually probing c is at least $\frac{y_c}{\alpha}(1 - \frac{k}{2\alpha})$. Finally, by linearity of expectations (since the instantiation of item c is independent of the event that it is probed) the expected profit is at least $\frac{1}{\alpha}(1 - \frac{k}{2\alpha}) \cdot \sum_{c=1}^{n} w_c \cdot y_c$. Setting $\alpha = k$ implies an expected profit of at least $\frac{1}{2k} \cdot \sum_c w_c y_c$, which proves the first part of Theorem 1.

2.1 Special Case: Monotone Column Outcomes

We now consider a special case of stochastic k-set packing where the outcomes of each column e form a total order w.r.t. the vector dominance relation; ie. for any column $i \in [n]$ and outcomes $a, b \in \{0,1\}^d$ for column i, either $a \leq b$ or $b \leq a$ coordinate-wise. Observe that this is true for the stochastic matching problem. The algorithm for monotone column outcomes is identical to the one for the general case when we set parameter $\alpha = 1$. We show below that this algorithm achieves a $k+1$ approximation; this bound nearly matches the LP integrality gap of $k - 1 + \frac{1}{k}$ for even deterministic k-set packing [13].

As above, consider the indicator random variables $\{\mathbf{I}_{c,\ell}\}_{\ell=1}^k$ for each column $c \in [n]$. The improvement for the monotone-outcome case comes from the following strengthened bound on $\Pr[\bigwedge_\ell(\neg\mathbf{I}_{c,\ell})]$ which is obtained via the FKG inequality ([2, Theorem 6.2.1]). Given a vector $X = \{X_1, \ldots, X_n\}$ of independent events and an event F which is a function of X, we say F is an *increasing* (decreasing) event if for any vector X that $F(X)$ holds, $F(Y)$ also holds when $Y_i \geq X_i\ \forall i$ ($X_i \geq Y_i\ \forall i$). The FKG inequality says that for any collection of increasing (decreasing) events $F_1, \ldots, F_k$, it holds that $\Pr[\bigwedge_{i=1}^k F_i] \geq \prod_{i=1}^k \Pr[F_i]$.

Lemma 2 *For any column $c \in [n]$, $\Pr[\bigwedge_\ell(\neg\mathbf{I}_{c,\ell})] \geq \frac{1}{k+1}$.*

Proof: We can assume that the random permutation π is chosen by the following random experiment: For each column e, we pick independently and uniformly at random a real number $a_e \in [0,1]$. The columns are then sorted in increasing order of these numbers to obtain π.

We first condition on $a_c = x$, and bound $\Pr[\bigwedge_\ell(\neg\mathbf{I}_{c,\ell})|a_c = x]$. For each column $e \in [n] \setminus \{c\}$, let the random variable $B_e = 1$ if $a_e \leq x$ and $B_e = 0$ otherwise. Let Z_e be the random variable corresponding to the random outcome of column e, with values consistent with the total-order of its outcomes. Let Y_e be the indicator random variable that is 1 w.p. y_e. Observe that random variables $\{B_e, Z_e, Y_e | e \in [n] \setminus \{c\}\}$ are mutually independent. Since the outcomes of each column e forms a total ordering, we can see that $\neg\mathbf{I}_{c,\ell}$ (for each $\ell \in [k]$) is a *decreasing function* of $\{B_e, Z_e, Y_e | e \in [n] \setminus \{c\}\}$. Therefore, by the FKG inequality, we have

$$\Pr\left[\bigwedge_\ell(\neg\mathbf{I}_{c,\ell}) \mid a_c = x\right] \geq \prod_\ell \Pr[(\neg\mathbf{I}_{c,\ell}) \mid a_c = x] \tag{3}$$

Claim 2 *For any column $c \in [n]$ and index $\ell \in [k]$, $\Pr[\mathbf{I}_{c,\ell} \mid a_c = x] \leq x$.*

Proof: Let $j \in [d]$ be the ℓ^{th} constraint in the support of c. Let U_c^j denote the usage of constraint j, when column c is considered (according to π). Then,

$$\begin{aligned}
\mathbb{E}[U_c^j \mid a_c = x] &= \sum_{e=1}^n \Pr[\, a_e < x \text{ AND } e \text{ is probed}\,] \cdot \mu_e(j), \\
&= \sum_{e=1}^n \Pr[\, a_e < x\,] \cdot y_e \cdot \mu_e(j), \\
&= \sum_{e=1}^n x \cdot y_e \cdot \mu_e(j), \\
&\leq x \cdot b_j.
\end{aligned}$$

Since $\mathbf{I}_{c,\ell} = \{U_c^j \geq b_j\}$, Markov's inequality implies that $\Pr[\mathbf{I}_{c,\ell} \mid a_c = x] \leq \frac{\mathbb{E}[U_c^j | a_c = x]}{b_j} \leq x$. ∎

Now using Inequality (3) and Claim 2, we have that

$$\begin{aligned}
\Pr\left[\bigwedge_\ell(\neg\mathbf{I}_{c,\ell})\right] &= \int_0^1 \Pr\left[\bigwedge_\ell(\neg\mathbf{I}_{c,\ell}) \mid a_c = x\right] \mathrm{d}x \geq \int_0^1 \prod_\ell \Pr[(\neg\mathbf{I}_{c,\ell}) \mid a_c = x]\,\mathrm{d}x \\
&\geq \int_0^1 \prod_\ell (1-x)\,\mathrm{d}x \geq \int_0^1 (1-x)^k\,\mathrm{d}x = \frac{1}{1+k}
\end{aligned}$$

This completes the proof of Lemma 2. ∎

Now, the probability of actually probing column c is at least $y_c \cdot \Pr[\wedge_\ell(\neg\mathbf{I}_{c,\ell})] \geq \frac{y_c}{k+1}$. Finally, by linearity of expectations (since the instantiation of item c is independent of the event that it is probed) the expected profit is at least $\frac{1}{k+1} \cdot \sum_{c=1}^n w_c \cdot y_c$. This proves the second part of Theorem 1.

2.2 Safe versus Unsafe policies

Here we show that our algorithm's policy (which is safe) achieves a good approximation even relative to the optimal unsafe policy. Recall that an item can be probed in a *safe* policy only if there is zero probability of violating any capacity constraint. Whereas an *unsafe* policy may probe an item even if there is positive probability of violating capacity—but if capacity is violated then no profit is received from that item and the policy ends. For a given set of items (with their distributions) and capacity vector $b' \in \mathbb{Z}_+^d$, let $\mathsf{Safe}(b')$ (resp. $\mathsf{Unsafe}(b')$) denote the value of the optimal safe (resp. unsafe) policy with capacity b'; $\mathsf{LP}(b')$ the optimal value of (LP1) with right hand side in (1) being b'; and $\mathsf{ALG}(b')$ the value obtained by our algorithm. Let $b \in \mathbb{Z}_+^d$ denote the capacity vector for the given instance; i.e. $b_j \geq 1$ for all $j \in [d]$ (if b_j was allowed to be 0 then clearly any safe policy gets zero value from items participating in this constraint j, but an unsafe policy can get positive value). We have:

$$\mathsf{Unsafe}(b) \leq \mathsf{Safe}(b + \mathbf{1}) \leq \mathsf{LP}(b + \mathbf{1}) \leq 2 \cdot \mathsf{LP}\left(\left\lceil \frac{b + \mathbf{1}}{2} \right\rceil\right) \leq 2 \cdot \mathsf{LP}(b)$$

where $\mathbf{1}$ is the all-ones vector. The first inequality uses the fact that each size lies in $\{0, 1\}$, the second is by Claim 1, the third is by scaling (since $\frac{b+\mathbf{1}}{2} \leq \lceil \frac{b+\mathbf{1}}{2} \rceil$), and the last inequality uses $b \in \mathbb{Z}_+^d$. Finally, the analysis in the previous subsections implies that $\mathsf{ALG}(b) \geq \frac{1}{2k} \cdot \mathsf{LP}(b)$ in general; and $\mathsf{ALG}(b) \geq \frac{1}{k+1} \cdot \mathsf{LP}(b)$ in case of monotone column-outcomes. Combined with the above inequality we have $\mathsf{ALG}(b) \geq \frac{1}{4k} \cdot \mathsf{Unsafe}(b)$, and $\mathsf{ALG}(b) \geq \frac{1}{2k+2} \cdot \mathsf{Unsafe}(b)$ in the monotone column-outcomes case.

3 Stochastic Matching

We consider the following stochastic matching problem. The input is an undirected graph $G = (V, E)$ with a weight w_e and a probability value p_e on each edge $e \in E$. In addition, there is an integer value t_v for each vertex $v \in V$ (called *patience parameter*). Initially, each vertex $v \in V$ has patience t_v. At each step in the algorithm, any edge $e(u, v)$ such that u and v have positive remaining patience can be probed. Upon probing edge e, one of the following happens: (1) with probability p_e, vertices u and v get *matched* and are removed from the graph (along with all adjacent edges), or (2) with probability $1 - p_e$, the edge e is removed and the remaining patience numbers of u and v get reduced by 1. An algorithm is an adaptive strategy for probing edges: its performance is measured by the expected weight of matched edges. The *unweighted* stochastic matching problem is the special case when all edge-weights are uniform.

Consider the following linear program: as usual, for any vertex $v \in V$, $\partial(v)$ denotes the edges incident to v. Variable y_e denotes the probability that edge $e = (u, v)$ gets probed in the adaptive strategy, and $x_e = p_e \cdot y_e$ denotes the probability that u and v get matched in the strategy. (This LP is similar to the LP used for general stochastic packing problems by Dean, Goemans and Vondrák [10].)

$$\text{maximize} \sum_{e \in E} w_e \cdot x_e \tag{LP2}$$

subject to

$$\sum_{e \in \partial(v)} x_e \leq 1 \qquad \forall v \in V \tag{4}$$

$$\sum_{e \in \partial(v)} y_e \leq t_i \qquad \forall v \in V \tag{5}$$

$$x_e = p_e \cdot y_e \qquad \forall e \in E \tag{6}$$

$$0 \leq y_e \leq 1 \qquad \forall e \in E \tag{7}$$

3.1 Weighted Stochastic Matching: Bipartite Graphs

In this section, we consider the stochastic matching problem on bipartite graphs. In fact, the algorithm produces a *matching-probing strategy* whose expected value is a constant fraction of the optimal value of (LP2) (which was for edge-probing).

Algorithm. First, we find an optimal fractional solution (x, y) to (LP2) and round y to identify a set of interesting edges $\widehat{E}$. Then we use König's Theorem [28, Ch. 20] to partition $\widehat{E}$ into a small collection of matchings $M_1, \ldots, M_h$. Finally, these matchings are then probed in random order. If we are only interested in edge-probing strategies, probing the edges in $\widehat{E}$ in random order would suffice. We will refer to this algorithm as ROUND-COLOR-PROBE:

1. $(x, y) \leftarrow$ optimal solution to (LP2)
2. $\widehat{y} \leftarrow$ round y to an integral solution using GKSP
3. $\widehat{E} \leftarrow \{e \in E : \widehat{y}_e = 1\}$
4. $M_1, \ldots, M_h \leftarrow$ optimal edge coloring of $\widehat{E}$
5. For each M in $\{M_1, \ldots, M_h\}$ in random order, do:
   a. probe $\{(u, v) \in M : u$ and v are unmatched$\}$

The algorithm above uses the GKSP procedure of Gandhi *et al.* [14], which we describe next.

The GKSP algorithm. We state some properties of the dependent rounding framework of Gandhi *et al.* [14] that are relevant in our context.

Theorem 7 ([14]) *Let $(A, B; E)$ be a bipartite graph and $z_e \in [0, 1]$ be fractional values for each edge $e \in E$. The GKSP algorithm is a polynomial-time randomized procedure that outputs values $Z_e \in \{0, 1\}$ for each $e \in E$ such that the following properties hold:*

P1. Marginal distribution. For every edge e, $\Pr[Z_e = 1] = z_e$.
P2. Degree preservation. For every vertex $u \in A \cup B$, $\sum_{e \in \partial(u)} Z_e \leq \lceil \sum_{e \in \partial u} z_e \rceil$.
P3. Negative correlation. For any vertex u and any set of edges $S \subseteq \partial(u)$:

$$\Pr[\bigwedge_{e \in S} (Z_e = 1)] \leq \prod_{e \in S} \Pr[Z_e = 1].$$

We note that the GKSP algorithm in fact guarantees stronger properties than the ones stated above. For the purpose of analyzing ROUND-COLOR-PROBE, however, the properties stated above will suffice.

Feasibility. Let us first argue that our algorithm outputs a feasible strategy. If we care about feasibility in the edge-probing model, we only need to show that each vertex u is not probed more than t_u times. The following lemma shows that:

Lemma 3 *For every vertex u,* ROUND-COLOR-PROBE *probes at most t_u edges incident on u.*

Proof: Vertex u is matched in $\left|\{e \in \partial_{\widehat{E}}(u)\}\right|$ matchings. This is an upper bound on the number of times edges incident on u probed. Hence we just need to show that this quantity is at most t_u. Indeed,

$$\left|\{e \in \partial_{\widehat{E}}(u)\}\right| = \sum_{e \in \partial(u)} \hat{y}_e \leq \left\lceil \sum_{e \in \partial(u)} y_e \right\rceil \leq t_u,$$

where the first inequality follows from the degree preservation property of Theorem 7 and the second inequality from the fact that y is a feasible solution to (LP2). ■

Let us argue that the strategy is also feasible under the matching-probing model. Recall that in the latter model we are given an additional parameter k (which without loss of generality we can assume to be at most $\max_{v \in V} t_u$) and we can probe edges in k round, with each round forming a matching. Let $\widehat{E}$ be the set of edges in the support of $\widehat{y}$, i.e., $\widehat{E} = \{e \in E \mid \widehat{y}_e = 1\}$. Let $h = \max_{v \in V} \deg_{\widehat{E}}(v) \leq \max_{v \in V} t_v$. König's Theorem allows us to decomposed $\widehat{E}$ into h matchings. Therefore, the probing strategy devised by the algorithm is also feasible in the matching-probing model.

Performance guarantee. Let us focus our attention on some edge $e = (u, v) \in E$. Our goal is to show that there is good chance that the algorithm will indeed probe e. We first analyze the probability of e being probed conditioned on $\widehat{E}$. Notice that the algorithm will probe e if and only if all previous probes incident on u and v were unsuccessful; otherwise, if there was a successful probe incident on u or v, we say that e was *blocked*.

Let π be a permutation of the matchings $M_1, \ldots, M_h$. We extend this ordering to the set $\widehat{E}$ by listing the edges within a matching in some arbitrary but fixed order. Let us denote by $B(e, \pi) \subseteq \widehat{E}$ the set of edges incident on u or v that appear before e in π. It is not hard to see that

$$\Pr\left[\, e \text{ was not blocked } \mid \widehat{E}\,\right] \geq \mathbb{E}_\pi\Big[\prod_{f \in B(e,\pi)} (1 - p_f) \mid \widehat{E} \Big]; \tag{8}$$

here we assume that $\prod_{f \in B(e,\pi)}(1 - p_f) = 1$ when $B(e, \pi) = \emptyset$.

Notice that in (8) we only care about the order of edges incident on u and v. Furthermore, the expectation does not range over all possible orderings of these edges, but only those that are consistent with some matching permutation. We call this type of restricted ordering *random matching ordering* and we denote it by π; similarly, we call an unrestricted ordering *random edge ordering* and we denote it by σ. Our plan is to study first the expectation in (8) over random edge orderings and then to show that the expectation can only increase when restricted to range over random matching orderings.

The following simple lemma is useful in several places.

Lemma 4 *Let r and p_{max} be positive real values. Consider the problem of minimizing $\prod_{i=1}^{t}(1 - p_i)$ subject to the constraints $\sum_{i=1}^{t} p_i \leq r$ and $0 \leq p_i \leq p_{\max}$ for $i = 1, \ldots, t$. Denote the minimum value by $\eta(r, p_{\max})$. Then,*

$$\eta(r, p_{\max}) = (1 - p_{\max})^{\lfloor \frac{r}{p_{\max}} \rfloor} \Big(1 - (r - \Big\lfloor \frac{r}{p_{\max}} \Big\rfloor p_{\max})\Big) \geq (1 - p_{\max})^{r/p_{\max}}.$$

Proof: Suppose the contrary that the quantity is minimized but there are two p_is that are strictly between 0 and $p_{\max}$. W.l.o.g, they are p_1, p_2 and $p_1 > p_2$ Let $\epsilon = \min(p_{\max} - p_1, p_2)$. It is easy to see that

$$(1 - (p_1 + \epsilon))(1 - (p_2 - \epsilon)) \prod_{i=3}^{t}(1 - p_i) - \prod_{i=1}^{t}(1 - p_i) = \epsilon(p_2 - p_1 - \epsilon) \prod_{i=3}^{t}(1 - p_i) < 0.$$

This contradicts the fact the quantity is minimized. Hence, there is at most one p_i which is strictly between 0 and $p_{\max}$.

The last inequality holds since $1 - b \geq (1 - a)^{b/a}$ for any $0 \leq b \leq a \leq 1$. ■

Let $\partial_{\widehat{E}}(e)$ be the set of edges in $\widehat{E}$ incident on either endpoint of e excluding e itself.

Lemma 5 *Let e be an edge in $\widehat{E}$ and let σ be a random edge ordering. Let $p_{\max} = \max_{f \in \widehat{E}} p_f$. Assume that $\sum_{f \in \partial_{\widehat{E}}(e)} p_f = r$. Then,*

$$\mathbb{E}_\sigma\Big[\prod_{f \in B(e,\sigma)} (1 - p_f) \mid \widehat{E} \Big] \geq \int_0^1 \eta(xr, xp_{\max}) \, \mathrm{d}x.$$

Proof: We claim that the expectation can be written in the following continuous form:

$$\mathbb{E}_\sigma\Big[\prod_{f \in B(e,\sigma)} (1 - p_f) \mid \widehat{E} \Big] = \int_0^1 \prod_{f \in \partial_{\widehat{E}}(e)} (1 - xp_f) \, \mathrm{d}x.$$

The lemma easily follows from this and Lemma 4.

To see the claim, we consider the following random experiment: For each edge $f \in \partial(e)$, we pick uniformly at random a real number a_f in $[0, 1]$. The edges are then sorted according to these numbers. It is not difficult to see that the experiment produces uniformly random orderings. For each edge f, let the random variable $A_f = 1 - p_f$ if $f \in B(e, \sigma)$ and $A_f = 1$ otherwise. Hence, we have

$$\mathbb{E}_\sigma\Big[\prod_{f \in B(e,\sigma)} (1 - p_f) \mid \widehat{E} \Big] = \int_0^1 \mathbb{E}\Big[\prod_{f \in \partial_{\widehat{E}}(e)} A_f \mid a_e = x \Big] \mathrm{d}x$$

$$= \int_0^1 \prod_{f \in \partial_{\widehat{E}}(e)} \mathbb{E}\Big[A_f \;\mid\; a_e = x \Big] \,\mathrm{d}x$$

$$= \int_0^1 \prod_{f \in \partial_{\widehat{E}}(e)} \big(x(1-p_f) + (1-x) \big) \,\mathrm{d}x$$

$$= \int_0^1 \prod_{f \in \partial_{\widehat{E}}(e)} (1 - x p_f) \,\mathrm{d}x$$

The second equality holds since the A_f variables, conditional on $a_e = x$, are independent. ∎

Lemma 6 *Let $\rho(r, p_{\max}) = \int_0^1 \eta(xr, xp_{\max}) \,\mathrm{d}x$. For any $r, p_{\max} > 0$, we have*

1. *$\rho\left(r, p_{\max}\right)$ is convex and decreasing on r.*
2. *$\rho\left(r, p_{\max}\right) \geq \frac{1}{r+p_{\max}} \cdot \left(1 - (1-p_{\max})^{1+\frac{r}{p_{\max}}}\right) > \frac{1}{r+p_{\max}} \cdot \left(1 - e^{-r}\right)$*

Proof: To see the first part, let us consider the function values on discrete points $r = p_{\max}, 2p_{\max}, \ldots$. Let $F(x) = \frac{1}{x}(1 - c^x)$ where $c = 1 - p_{\max}$. From Lemma 4, we can easily get that for integral t,

$$\rho\left(tp_{\max}, p_{\max}\right) = \int_0^1 (1 - xp_{\max})^t \,\mathrm{d}x = \frac{1}{p_{\max}(t+1)} \left(1 - c^{t+1}\right) = \frac{1}{p_{\max}} F(t+1).$$

The function $F(x)$ is a convex function for any $0 < c < 1$. Indeed, it is not hard to prove that $\frac{\mathrm{d}^2}{\mathrm{d}x^2} F(x) = \frac{2}{x^3} + c^x \left(-\frac{2}{x^3} + \frac{2 \ln a}{x^2} - \frac{\ln^2 a}{x} \right) > 0$ for any $0 < c < 1$. However, $\rho\left(tp_{\max}, p_{\max}\right)$ only coincides with $\frac{1}{p_{\max}} F(t+1)$ at integral values of t. Now, let us consider the value of $\rho(r, p_{\max})$ for $\gamma p_{\max} < r < (\gamma+1)p_{\max}$ (for some integer $\gamma \geq 0$):

$$\rho\left(r, p_{\max}\right) = \int_0^1 (1 - xp_{\max})^{\gamma} \Big(1 - x(r - \gamma p_{\max})\Big) \,\mathrm{d}x \tag{9}$$

The key observation is that for fixed values of $p_{\max}$ and γ the right hand side of (9) is a just linear function of r. The dependency of ρ in terms of r then becomes clear: it is a piecewise linear function that takes the value $F(t+1)/p_{\max}$ at abscissa points $tp_{\max}$ for $t \in \mathbb{Z}_{\geq 0}$. Therefore, ρ is a convex decreasing function of r.

The second part follows easily from Lemma 4:

$$\rho\left(r, p_{\max}\right) = \int_0^1 \eta(xr, xp_{\max}) \,\mathrm{d}x \geq \int_0^1 (1 - xp_{\max})^{r/p_{\max}} \,\mathrm{d}x$$
$$= \frac{1}{r + p_{\max}} \cdot \left(1 - (1-p_{\max})^{1+\frac{r}{p_{\max}}}\right) \geq \frac{1}{r + p_{\max}} \cdot \left(1 - e^{-r}\right)$$

∎

Lemma 7 *Let $e = (u, v) \in \widehat{E}$. Let π be a random matching ordering and σ be a random edge ordering of the edges adjacent to u and v. Then*

$$\mathbb{E}_{\pi}\Big[\prod_{f \in B(e,\pi)} (1 - p_f) \;\mid\; \widehat{E} \Big] \geq \mathbb{E}_{\sigma}\Big[\prod_{f \in B(e,\sigma)} (1 - p_f) \;\mid\; \widehat{E} \Big].$$

Proof: We can think of π as a permutation of bundles of edges: For each matching, if there are two edges incident on e, we bundle the edges together; if there is a single edge incident on e this edge is in a singleton bundle by itself. The random edge ordering σ can be thought as having all edges incident on e in singleton bundles.

Consider the same random experiment as in Lemma 5 except that we only pick one random number for each bundle. Let $G(e)$ be the set of all bundles incident on e. Using the same argument as in Lemma 5, we have

$$\mathbb{E}_{\pi}\Big[\prod_{f \in B(e,\pi)} (1 - p_f) \;\mid\; \widehat{E} \Big] = \int_0^1 \prod_{g \in G(e)} \Big(x \cdot \prod_{f \in g} (1 - p_f) + (1 - x) \Big) \,\mathrm{d}x.$$

But for any bundle $g \in G(e)$ and $0 \le x \le 1$, we claim that

$$x \cdot \prod_{f \in g} (1 - p_f) + (1 - x) \ge \prod_{f \in g} (1 - x p_f).$$

For singleton bundles we actually have equality. For a bundle $g = \{f_1, f_2\}$, we have $x(1-p_{f_1})(1-p_{f_2})+(1-x) = 1 - xp_{f_1} - xp_{f_2} + xp_{f_1}p_{f_2} \ge 1 - xp_{f_1} - xp_{f_2} + x^2 p_{f_1}p_{f_2} = (1 - xp_{f_1})(1 - xp_{f_2})$. This completes the proof. ■

As we shall see shortly, if $\sum_{f \in \partial_{\widehat{E}}(e)} p_e$ is small then the probability that e is not blocked is large. Because of the marginal distribution property of the GKSP rounding procedure, we can argue that this quantity is small *in expectation* since $\sum_{f \in \partial(e)} p_e y_e \le 2$ due to the fact that y is a feasible solution to (LP2). This, however, is not enough; in fact, for our analysis to go through, we need a slightly stronger property.

Lemma 8 *For every edge e,*

$$\mathbb{E}\Big[\sum_{f \in \partial_{\widehat{E}}(e)} p_f \;\mid\; e \in \widehat{E} \Big] \le \sum_{f \in \partial(e)} p_f \, y_f.$$

Proof: Let u be an endpoint of e.

$$\begin{aligned}
\mathbb{E}\Big[\sum_{f \in \partial_{\widehat{E}}(u) - e} p_f \;\mid\; e \in \widehat{E} \Big] &= \sum_{f \in \partial(u) - e} \Pr[\,\widehat{y}_f = 1 \;\mid\; \widehat{y}_e = 1] \cdot p_f, \\
&\le \sum_{f \in \partial(u) - e} \Pr[\,\widehat{y}_f = 1\,] \cdot p_f, && \text{[by Theorem 7 P3]} \\
&= \sum_{f \in \partial(u) - e} y_f \, p_f. && \text{[by Theorem 7 P1].}
\end{aligned}$$

The same bound holds for the other endpoint of e. Adding the two inequalities we get the lemma. ■

Everything is in place to derive a bound the expected weight of the matching found by our algorithm.

Theorem 8 *If G is bipartite then* ROUND-COLOR-PROBE *is a $1/\rho(2, p_{\max})$ approximation under the edge- and matching-probing model, where ρ is defined in Lemma 6. The worst ratio is attained at $p_{\max} = 1$, where it is 3. The ratio tends to $\frac{2}{1-e^{-2}}$ as $p_{\max}$ tends to 0.*

Proof: Recall that the optimal value of (LP2) is exactly $\sum_{e \in E} w_e y_e x_e$. The expected weight of the matching found by the algorithm is

$$\begin{aligned}
\mathbb{E}\,[\,\mathsf{ALG}\,] &= \sum_{e \in E} w_e \, p_e \, \Pr[\, e \in \widehat{E}\,] \cdot \Pr\,[\, e \text{ was not blocked } \mid e \in \widehat{E}\,] \\
&= \sum_{e \in E} w_e \, p_e \, y_e \cdot \Pr\,[\, e \text{ was not blocked } \mid e \in \widehat{E}\,] && \text{[by Theorem 7 P1]} \\
&\ge \sum_{e \in E} w_e \, p_e \, y_e \cdot \mathbb{E}_\pi \Big[\prod_{f \in B(e,\pi)} (1 - p_f) \;\mid\; e \in \widehat{E} \Big] && \text{[by (8)]} \\
&\ge \sum_{e \in E} w_e \, p_e \, y_e \cdot \mathbb{E}_\sigma \Big[\prod_{f \in B(e,\sigma)} (1 - p_f) \;\mid\; e \in \widehat{E} \Big] && \text{[by Lemma 7]} \\
&\ge \sum_{e \in E} w_e \, p_e \, y_e \cdot \mathbb{E}\Big[\rho\Big(\sum_{f \in \partial_{\widehat{E}}(e)} p_f, p_{\max} \Big) \;\mid\; e \in \widehat{E} \Big] && \text{[by Lemma 5]} \\
&\ge \sum_{e \in E} w_e \, p_e \, y_e \cdot \rho\Big(\mathbb{E}\Big[\sum_{f \in \partial_{\widehat{E}}(e)} p_f \;\mid\; e \in \widehat{E} \Big], p_{\max} \Big) && \text{[by Jensen's inequality]} \\
&\ge \sum_{e \in E} w_e \, p_e \, y_e \cdot \rho\Big(\sum_{f \in \partial(e)} y_f \, p_f, p_{\max} \Big) && \text{[by Lemma 8]} \\
&\ge \sum_{e \in E} w_e \, p_e \, y_e \cdot \rho(2, p_{\max}) && \text{[y is feasible for (LP2)].}
\end{aligned}$$

Notice that we are able to use Jensen's inequality because, as shown in Lemma 6, $\rho(r, p_{\max})$ is a convex and decreasing function of r. The last two inequalities also use the fact that ρ is decreasing.

It can be checked directly that $\rho(2, p_{max})$ is maximized at $p_{max} = 1$ where it is 3. Moreover $\rho(2, p_{max}) \to \left(1 - e^{-2}\right)/2$ as p_{max} tends to 0. ■

This proves the second parts of Theorem 2 and Theorem 3.

3.2 Weighted Stochastic Matching: General Graphs

We now present an algorithm for weighted stochastic matching in general graphs that builds on the algorithm for the bipartite case. The basic idea is to solve (LP2), randomly partition the vertices of G into two sets A and B, and then run ROUND-COLOR-PROBE on the bipartite graph induced by (A, B). For the analysis to go through, it is crucial that we use the already computed fractional solution instead of solving again (LP2) for the new bipartite graph in the call to ROUND-COLOR-PROBE.

1. $(x, y) \leftarrow$ optimal solution to (LP2)
2. randomly partition vertices into A and B
3. run ROUND-COLOR-PROBE on the bipartite graph and the fractional solution induced by (A, B)

Theorem 9 *For general graphs there is a $2/\rho(1, p_{\max})$ approximation under the edge- and matching-probing model, where ρ is defined in Lemma 6. The worst ratio is attained at $p_{\max} = 1$, where it is 4. The ratio tends to $\frac{2}{1-e^{-1}}$ as $p_{\max}$ tends to 0.*

Proof: The analysis is very similar to the bipartite case. Essentially, conditional on a particular outcome for the partition (A, B), all the lemmas derived in the previous section hold. In other words, the same derivation done in the proof of Theorem 8 yields:

$$\mathbb{E}[\,\mathsf{ALG} \mid (A, B)\,] \geq \sum_{e \in (A,B)} w_e p_e y_e \cdot \rho\Big(\sum_{f \in \partial_{A,B}(e)} p_f\, y_f, p_{\max} \Big),$$

where $\partial_{A,B}(e) = \partial(e) \cap (A, B)$.

Hence, the expectation of algorithm's performance is:

$$\begin{aligned}
\mathbb{E}[\,\mathsf{ALG}\,] &\geq \sum_{e \in E} w_e\, p_e\, y_e\, \Pr[e \in (A, B)] \cdot \mathbb{E}\Big[\rho\Big(\sum_{f \in \partial_{A,B}(e)} p_f\, y_f,\, p_{\max} \Big) \mid e \in (A, B) \Big], \\
&\geq \sum_{e \in E} w_e\, p_e\, y_e\, \frac{1}{2} \cdot \rho\Big(\mathbb{E}\Big[\sum_{f \in \partial_{A,B}(e)} p_f\, y_f \mid e \in (A, B) \Big],\, p_{\max} \Big), \\
&\geq \sum_{e \in E} w_e\, p_e\, y_e\, \frac{1}{2} \cdot \rho\Big(\sum_{f \in \partial(e)} \frac{p_f\, y_f}{2},\, p_{\max} \Big), \\
&\geq \sum_{e \in E} w_e\, p_e\, y_e\, \frac{1}{2} \cdot \rho\big(1,\, p_{\max}\big),
\end{aligned}$$

where the second inequality follows from Jensen's inequality and the fact that $\rho(r, p_{\max})$ is a convex decreasing function of r. Finally, noting that $\sum_{e \in E} w_e\, p_e\, y_e$ is a lower bound on the value of the optimal strategy, the theorem follows. ■

This proves the first parts of Theorem 2 and Theorem 3.

4 Stochastic *Online* Matching with Timeouts

As mentioned in the introduction, the stochastic online matching with timeouts is best imagined as selling a finite set of goods to buyers that arrive over time. The input to the problem consists of a bipartite graph $G = (A, B, A \times B)$, where A is the set of *items* that the seller has to offer, with exactly one copy of each item, and B is a set of *buyer types/profiles*. For each buyer type $b \in B$ and item $a \in A$, p_{ab} denotes the probability that a buyer of type b will like item a, and w_{ab} denotes the revenue obtained if item a is sold to a buyer of type b. Each buyer of type $b \in B$ also has a patience parameter $t_b \in \mathbb{Z}_+$. There are n buyers arriving online, with $e_b \in \mathbb{Z}$ denoting the expected number of buyers of type b, with $\sum e_b = n$. Let $\mathcal{D}$ denote the induced probability distribution on B by defining $\Pr_{\mathcal{D}}[b] = e_b/n$. All the above information is given as input.

The stochastic online model is the following: At each point in time, a buyer arrives, where her type $\mathbf{b} \in_{\mathcal{D}} B$ is an i.i.d. draw from $\mathcal{D}$. The algorithm now shows her *up to* $t_{\mathbf{b}}$ *distinct items one-by-one*: the buyer likes each item $a \in A$ shown to her independently with probability p_{ab}. The buyer purchases the first item that she is *offered and likes*; if she buys item a, the revenue accrued is w_{ab}. If she does not like any of the items shown, she leaves without buying. The objective is to maximize the expected revenue.

We get the stochastic online matching problem of Feldman *et al.* [12] if we have $w_{ab} = p_{ab} \in \{0, 1\}$, in which case we need only consider $t_b = 1$. Their focus was on beating the $1 - 1/e$-competitiveness known for worst-case models [23,22,27,7,15]; they gave a 0.67-competitive algorithm that works for the unweighted case with high probability. On the other hand, our results are for the weighted case (with preference-uncertainty and timeouts), but only in expectation. Furthermore, in our extension, due to the presence of timeouts (see Section 1.2), any algorithm that provides a guarantee whp must necessarily have a high competitive ratio.

By making copies of buyer types, we may assume that $e_b = 1$ *for all* $b \in B$*, and* $\mathcal{D}$ *is uniform over* B. For a particular run of the algorithm, let $\hat{B}$ denote the actual set of buyers that arrive during that run. Let $\hat{G} = (A, \hat{B}, A \times \hat{B})$, where for each $a \in A$ and $\hat{b} \in \hat{B}$ (and suppose its type is some $b \in B$), the probability associated with edge $(a, \hat{b})$ is p_{ab} and its weight is w_{ab}. Moreover, for each $\hat{b} \in \hat{B}$ (with type, say, $b \in B$), set its patience parameter to $t_{\hat{b}} = t_b$. We will call this the *instance graph*; the algorithm sees the vertices of $\hat{B}$ in random order, and has to adaptively find a large matching in $\hat{G}$.

It now seems reasonable that the algorithm of Section 2 (specialized to stochastic matching) should work here. But the algorithm does not know $\hat{G}$ (the actual instantiation of the buyers) up front, it only knows G, and hence some more work is required to obtain an algorithm. Further, as was mentioned in the preliminaries, we use OPT to denote the optimal adaptive strategy (instead of the optimal offline matching in $\hat{G}$ as was done in [12]), and compare our algorithm's performance with this OPT.

The Linear Program. For a graph $H = (A, C, A \times C)$ with each edge (a, c) having a probability p_{ac} and weight w_{ac}, and vertices in C having patience parameters t_j, consider the LP(H):

$$\text{maximize} \sum_{a \in A,\, c \in C} w_{ac} \cdot x_{ac} \tag{LP3}$$

subject to

$$\sum_{c \in C} x_{ac} \le 1 \qquad \forall a \in A \tag{10}$$

$$\sum_{a \in A} x_{ac} \le 1 \qquad \forall c \in C \tag{11}$$

$$\sum_{a \in A} y_{ac} \le t_c \qquad \forall c \in C \tag{12}$$

$$x_{ac} = p_{ac} \cdot y_{ac} \qquad \forall a \in A,\, c \in C \tag{13}$$

$$y_{ac} \in [0, 1] \qquad \forall a \in A,\, c \in C \tag{14}$$

Note that this LP is very similar to the one in Section 3, but the vertices in A do not have timeout values. Let $\mathsf{LP}(H)$ denote the optimal value of this LP.

The algorithm:

1. Before any buyers arrive, solve the LP on the expected graph G to get values y^*.
2. When any buyer $\hat{b}$ (of type b) arrives online:
   a. If $\hat{b}$ is the first buyer of type b, consider the items $a \in A$ in u.a.r. order. One by one, offer each item a (that is still unsold) to $\hat{b}$ independently with probability y^*_{ab}/α; stop if either t_b offers are made or $\hat{b}$ purchases any item.
   b. If $\hat{b}$ is not the first arrival of type b, do not offer any items to $\hat{b}$.

In the following, we prove that our algorithm achieves a constant approximation to stochastic online matching with timeouts. The first lemma show that the expected value obtained by the best online adaptive algorithm is bounded above by $\mathbb{E}[\mathsf{LP}(\hat{G})]$.

Lemma 9 *The optimal value* OPT *of the given instance is at most* $\mathbb{E}[\mathsf{LP}(\hat{G})]$*, where the expectation is over the random draws to create* $\hat{G}$*.*

Proof: Consider an algorithm that is allowed to see the instantiation $\hat{B}$ of the buyers before deciding on the selling strategy—the expected revenue of the best such algorithm is clearly an upper bound on OPT. Given any instantiation $\hat{B}$, the expected revenue of the optimal selling strategy is at most $\mathsf{LP}(\hat{G})$ (see e.g. Claim 1). The claim follows by taking an expectation over $\hat{B}$. ∎

Lemma 10 *Our expected revenue is at least* $\left(1-\frac{1}{e}\right)\frac{1}{\alpha}\left(1-\frac{1}{\alpha}-\frac{2}{3\alpha^2}\right)\cdot\mathsf{LP}(G)$*.*

Proof: For any buyer-type $b \in B$, in this proof, $\hat{b}$ refers to the first type-b buyer (if any). For each $b \in B$, let r.v. $T_b \in [n] \cup \{\infty\}$ denote the earliest arrival time of a type-b buyer; if there is no type-b arrival then $T_b = \infty$. Note that our algorithm obtains positive revenue only for buyers $\{\hat{b} \mid b \in B,\, T_b < \infty\}$; let R_b denote the revenue obtained from buyer $\hat{b}$ (if any). The expected revenue of the algorithm is $\mathbb{E}[\sum_{b\in B} R_b]$. We now estimate $\mathbb{E}[R_b]$ for a fixed $b \in B$.

Let $\mathcal{A}_b \equiv (T_b < \infty)$ denote the event that there is some type-b arrival in the instantiation $\hat{B}$. Since each arrival is i.i.d. from the uniform distribution over B, $\Pr[\mathcal{A}_b] = 1-(1-1/n)^n \geq 1-\frac{1}{e}$. In the following, we condition on $\mathcal{A}_b$ and bound $\mathbb{E}[R_b \mid \mathcal{A}_b]$. Hence we assume that buyer $\hat{b}$ exists.

For any vertex $a \in A$, let M_a denote the indicator r.v. that a is already matched before time T_b; and O_a (resp. M'_a) the indicator r.v. that $\hat{b}$ is timed-out (resp. already matched) when item a is considered for offering to $\hat{b}$. Now,

$$\begin{aligned}\Pr[\,\text{item } a \text{ offered to } \hat{b} \mid \mathcal{A}_b\,] &= (1-\Pr[\,M_a \cup M'_a \cup O_a \mid \mathcal{A}_b\,])\cdot\frac{y_{ab}}{\alpha}\\ &\geq (1-\Pr[\,M_a \mid \mathcal{A}_b\,]-\Pr[\,M'_a\cup O_a \mid \mathcal{A}_b\,])\cdot\frac{y_{ab}}{\alpha} \qquad (15)\end{aligned}$$

Claim 3 *For any* $a \in A$ *and* $b \in B$*,* $\Pr[\,M_a \mid \mathcal{A}_b\,] \leq \frac{1}{2\alpha}$*.*

Proof: For any $v \in B\setminus\{b\}$, let I^v_b denote the indicator r.v. for the event $T_v < T_b$. We have:

$$\begin{aligned}\Pr[\,M_a \mid \mathcal{A}_b\,] &= \sum_{v\in B\setminus\{b\}} \Pr[\,\text{type-}v \text{ buyer is matched to } a \text{ before time } T_b \mid \mathcal{A}_b\,] &(16)\\ &= \sum_{v\in B\setminus\{b\}} \Pr[\,I^v_b \mid \mathcal{A}_b\,]\cdot\Pr[\,\hat{v} \text{ matched to } a \mid I^v_b,\, \mathcal{A}_b\,] &(17)\\ &\leq \sum_{v\in B\setminus\{b\}} \Pr[\,I^v_b \mid \mathcal{A}_b\,]\cdot\frac{x_{av}}{\alpha} \quad\leq\quad \frac{1}{2}\sum_{v\in B\setminus\{b\}}\frac{x_{av}}{\alpha} \quad\leq\quad \frac{1}{2\alpha}, &(18)\end{aligned}$$

where the first inequality follows from the fact that even after the algorithm has considered an edge (a, v), the probability of matching (a, v) is $\frac{y_{av}}{\alpha}\cdot p_{av}$, the last inequality uses LP-constraint (10) for graph G, and the second last inequality uses $\Pr[\,I^v_b \mid \mathcal{A}_b\,] \leq \frac{1}{2}$ (for $v \in B\setminus\{b\}$), which we show next.

Note that event $I^v_b \wedge \mathcal{A}_b$ corresponds to $(T_v < T_b < \infty)$; and event $\mathcal{A}_b$ contains both $(T_v < T_b < \infty)$ and $(T_b < T_v < \infty)$. By symmetry, $\Pr[\,T_v < T_b < \infty\,] = \Pr[\,T_b < T_v < \infty\,]$, which implies:

$$\Pr[\,I^v_b \mid \mathcal{A}_b\,] = \frac{\Pr[\,T_v < T_b < \infty\,]}{\Pr[\mathcal{A}_b]} \leq \frac{\Pr[\,T_v < T_b < \infty\,]}{\Pr[\,(T_v < T_b < \infty)\vee(T_b < T_v < \infty)\,]} = \frac{1}{2}.$$

This completes the proof of Claim 3. ∎

Claim 4 *For any* $a \in A$ *and* $b \in B$*,* $\Pr[\,M'_a \cup O_a \mid \mathcal{A}_b\,] \leq \frac{1}{2\alpha}+\frac{2}{3\alpha^2}$*.*

Proof: It is easy to prove an upper bound of $\frac{1}{\alpha}$ via Markov's inequality. Since items offered to $\hat{b}$ are considered in u.a.r. order, we obtain $\Pr[\,O_a \mid \mathcal{A}_b\,] \leq \frac{1}{2\alpha}$ and $\Pr[\,M'_a \mid \mathcal{A}_b\,] \leq \frac{1}{2\alpha}$, using LP-constraints (12) and (11) respectively. This suffices to get a weaker constant approximation in Theorem 4.

Proving the stronger bound claimed above requires some case analysis:

- Suppose $t_b = 1$. Here we have $M'_a \subseteq O_a$, so $\Pr[\,M'_a \cup O_a \mid \mathcal{A}_b\,] = \Pr[\,O_a \mid \mathcal{A}_b\,] \leq \frac{1}{2\alpha}$.
- Suppose $t_b \geq 2$. In this case we prove $\Pr[\,O_a \mid \mathcal{A}_b\,] \leq \frac{2}{3\alpha^2}$ using a Chernoff-type bound (see Claim 5 below). Now by union bound, $\Pr[\,M'_a \cup O_a \mid \mathcal{A}_b\,] \leq \frac{1}{2\alpha}+\frac{2}{3\alpha^2}$.

In both cases above, the statement in Claim 4 holds. ■

Claim 5 *Suppose $\alpha \geq \mathrm{e}$. For any $b \in B$ with $t_b \geq 2$, and any $a \in A$, the probability that $\hat{b}$ has timed out when a is considered for offering to $\hat{b}$ is at most $\frac{2}{3\alpha^2}$.*

Proof: Let $t = t_b \geq 2$ denote the patience parameter for vertex b, and π the random order in which items A are considered. Let $F = A \setminus \{a\}$. Then the probability that $\hat{b}$ has timed out when a is considered is upper bounded by:

$$\sum_{\{p_1,\cdots,p_t\}\subseteq F} \Pr[\,\text{items } p_1,\cdots,p_t \text{ appear before } a \text{ in } \pi \text{ AND are all offered}\,], \tag{19}$$

$$\leq \frac{1}{t!}\cdot \sum_{p_1,\cdots,p_t\in F} \Pr[\,\text{items } p_1,\cdots,p_t \text{ appear before } f \text{ in } \pi \text{ AND are all offered}\,], \tag{20}$$

$$\leq \frac{1}{t!}\cdot \sum_{p_1,\cdots,p_t\in F} \Pr[\,\text{items } p_1,\cdots,p_t \text{ appear before } f \text{ in } \pi\,]\cdot \prod_{\ell=1}^{t}\frac{y_{p_\ell}}{\alpha}, \tag{21}$$

$$= \frac{1}{(t+1)!}\cdot \sum_{p_1,\cdots,p_t\in F} \prod_{\ell=1}^{t}\frac{y_{p_\ell}}{\alpha}, \tag{22}$$

$$= \frac{1}{(t+1)!}\cdot \left(\sum_{p\in F}\frac{y_p}{\alpha}\right)^t \tag{23}$$

$$\leq \frac{1}{(t+1)!}\cdot \left(\frac{t}{\alpha}\right)^t. \tag{24}$$

In the above, the summation in (19) is over unordered t-tuples whereas the subsequent ones (20)-(22) are over ordered tuples (with repetition). Inequality (21) uses the fact that for any item g, the probability of offering g *conditioned* on π and the outcomes until g is considered, is at most y_g/α. Equation (22) follows from the fact that probability that f is the last to appear among $\{p_1,\cdots,p_t,f\}$ in a random permutation π is $\frac{1}{t+1}$. Finally, (24) follows from the LP constraint (12) at b.

Let $f(t) := \frac{1}{(t+1)!}\cdot\left(\frac{t}{\alpha}\right)^t$. We claim that $f(t) \leq \frac{2}{3\alpha^2}$ when $\alpha \geq \mathrm{e}$ and $t \geq 2$, which would prove the claim. Note that this is indeed true for $t = 2$ (in fact with equality). Also $f(t+1) \leq f(t)$ for all $t \geq 2$ due to:

$$\frac{f(t+1)}{f(t)} = \left(\frac{t+1}{t}\right)^t \cdot \frac{t+1}{t+2}\cdot\frac{1}{\alpha} \leq \frac{\mathrm{e}}{\alpha} \leq 1.$$

Thus we obtain the desired upper bound. ■

Now applying Claims 3 and 4 to (15), we obtain:

$$\Pr[\,\text{item } a \text{ offered to } \hat{b} \mid \mathcal{A}_b\,] \geq \frac{1}{\alpha}\left(1-\frac{1}{\alpha}-\frac{2}{3\alpha^2}\right)\cdot y_{ab}.$$

This implies:

$$\begin{aligned}\mathbb{E}[R_b \mid \mathcal{A}_b] &= \sum_{a\in A} w_{ab}\cdot p_{ab}\cdot \Pr[\,\text{item } a \text{ offered to } \hat{b} \mid \mathcal{A}_b\,]\\ &\geq \frac{1}{\alpha}\left(1-\frac{1}{\alpha}-\frac{2}{3\alpha^2}\right)\sum_{a\in A} w_{ab}\cdot x_{ab}.\end{aligned}$$

Since $\Pr[\,\mathcal{A}_b\,] \geq 1-\frac{1}{e}$, we also have $\mathbb{E}[R_b] \geq (1-\frac{1}{e})\,\frac{1}{\alpha}\,\left(1-\frac{1}{\alpha}-\frac{2}{3\alpha^2}\right)\sum_{a\in A} w_{ab}\cdot x_{ab}$.

Finally, the expected revenue obtained by the algorithm is:

$$\sum_{b\in B}\mathbb{E}[R_b] \geq \left(1-\frac{1}{e}\right)\frac{1}{\alpha}\left(1-\frac{1}{\alpha}-\frac{2}{3\alpha^2}\right)\cdot \mathsf{LP}(G).$$

This proves Lemma 10. ■

Note that we have shown that $\mathbb{E}[\mathsf{LP}(\hat{G})]$ is an upper bound on OPT, and that we can get a constant fraction of $\mathsf{LP}(G)$. The final lemma relates these two, namely the LP-value of the expected graph G (computed in Step 1) to the expected LP-value of the instantiation $\hat{G}$; the proof uses a simple but subtle duality-based argument.

Lemma 11 $\mathsf{LP}(G) \geq \mathbb{E}[\mathsf{LP}(\hat{G})]$.

Proof: Consider the dual of the linear program (LP3).

$$\min \sum_{a\in A} \alpha_a + \sum_{c\in C} (\alpha_c + t_c \cdot \beta_c) + \sum_{a\in A,\, c\in C} z_{ac} \tag{25}$$

$$z_{ac} + p_{ac} \cdot (\alpha_a + \alpha_c) + \beta_c \geq w_{ac} \cdot p_{ac} \qquad \forall a \in A,\, c \in C \tag{26}$$

$$\alpha, \beta, z \geq 0 \tag{27}$$

Let (α, β, z) denote the optimal dual solution corresponding to graph G; note that its objective value equals $\mathsf{LP}(G)$ by strong duality. For any instantiation $\hat{G}$, define dual solution $(\hat{\alpha}, \hat{\beta}, \hat{z})$ as follows:

- For all $a \in A$, $\hat{\alpha}_a = \alpha_a$.
- For each $c \in \hat{B}$ (of type b), $\hat{\alpha}_c = \alpha_b$ and $\hat{\beta}_c = \beta_b$.
- For each $a \in A$ and $c \in \hat{B}$ (of type b), $\hat{z}_{ac} = z_{ab}$.

Note that $(\hat{\alpha}, \hat{\beta}, \hat{z})$ is a feasible dual solution corresponding to the LP on $\hat{G}$: there is constraint for each $a \in A$ and $c \in \hat{B}$, which reduces to a constraint for (α, β, z) in the dual corresponding to G. By weak duality, the objective value for $(\hat{\alpha}, \hat{\beta}, \hat{z})$ is an upper-bound on $\mathsf{LP}(\hat{G})$. For each $b \in B$, let N_b denote the number of type b buyers in the instantiation $\hat{B}$; note that $\mathbb{E}[N_b] = 1$ by definition of distribution $\mathcal{D}$. Then the dual objective for $(\hat{\alpha}, \hat{\beta}, \hat{z})$ satisfies:

$$\sum_{a\in A} \alpha_a + \sum_{b\in B} N_b \cdot (\alpha_b + t_b \cdot \beta_b) + \sum_{a\in A,\, b\in B} N_b \cdot z_{ab} \quad \geq \quad \mathsf{LP}(\hat{G}).$$

Taking an expectation over $\hat{B}$, we obtain:

$$\begin{aligned} \mathbb{E}[\mathsf{LP}(\hat{G})] &\leq \sum_{a\in A} \alpha_a + \sum_{b\in B} \mathbb{E}[N_b] \cdot \left(\alpha_b + t_b \cdot \beta_b + \sum_{a\in A} z_{ab} \right) \\ &= \sum_{a\in A} \alpha_a + \sum_{b\in B} (\alpha_b + t_b \cdot \beta_b) + \sum_{a\in A,\, b\in B} z_{ab} = \mathsf{LP}(G). \end{aligned}$$

This proves the lemma. ■

Applying Lemmas 9, 10 and 11, and setting $\alpha = \frac{2}{\sqrt{3}-1}$, completes Theorem 4's proof.

5 Final Remarks

An extended abstract of this paper appeared in the Proceedings of the 18th Annual European Symposium on Algorithms [4]. The bounds presented here in Section 3 are slightly better than those claimed in the extended abstract. It remains an open question to obtain nontrivial lower bounds for the stochastic matching problem.

Acknowledgements. We would like to thank Aravind Srinivasan for helpful discussions.

References

1. M. Adamczyk. Greedy algorithm for stochastic matching is a 2-approximation, 2010. arXiv:1007.3036.
2. N. Alon and J. H. Spencer. *The probabilistic method*. Wiley-Interscience, 2008.
3. B. Bahmani and M. Kapralov. Improved bounds for online stochastic matching. In *Proceedings of the 18th European Symposium on Algorithms, LNCS 6346*, pages 170–181. Springer, 2010.
4. N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When LP is the Cure for Your Matching Woes: Improved Bounds for Stochastic Matchings. In *Proceedings of the 18th European Symposium on Algorithms, LNCS 6346*, pages 218–229. Springer, 2010.
5. N. Bansal, N. Korula, V. Nagarajan, and A. Srinivasan. On k-column sparse packing programs. *Proceedings of the 14th Conference on Integer Programming and Combinatorial Optimization, LNCS 6080*, pages 369–382, 2010.
6. S. Bhattacharya, G. Goel, S. Gollapudi, and K. Munagala. Budget constrained auctions with heterogeneous items. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, pages 379–388. ACM, 2010.
7. B. E. Birnbaum and C. Mathieu. On-line bipartite matching made simple. *SIGACT News*, 39(1):80–87, 2008.

8. R. Carr and S. Vempala. Randomized metarounding. *Random Structures Algorithms*, 20(3):343–352, 2002. Probabilistic methods in combinatorial optimization.
9. N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian, and A. Rudra. Approximating matches made in heaven. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming, LNCS 5555*, pages 266–278, 2009.
10. B. C. Dean, M. X. Goemans, and J. Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings fo the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 395–404, 2005.
11. B. C. Dean, M. X. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: the benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964, 2008.
12. J. Feldman, A. Mehta, V. S. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating $1 - 1/e$. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2009.
13. Z. Füredi, J. Kahn, and P. Seymour. On the Fractional Matching Polytope of a Hypergraph. *Combinatorica*, 13(2):167–180, 1993.
14. R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):360, 2006.
15. G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 982–991, 2008.
16. S. Guha and K. Munagala. Approximation algorithms for partial-information based stochastic control with markovian rewards. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 483–493, 2007.
17. S. Guha and K. Munagala. Multi-armed bandits with metric switching costs. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming, LNCS 5555*, pages 496–507, 2009.
18. A. Gupta, R. Krishnaswamy, M. Molinaro, and R. Ravi. Approximation algorithms for correlated knapsacks and non-martingale bandits, 2011. CoRR, abs/1102.3749.
19. A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 417–426. ACM, 2004.
20. E. Hazan, S. Safra, and O. Schwartz. On the complexity of approximating k-set packing. *Computational Complexity*, 15(1):20–39, 2006.
21. C. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics*, 2(1):68–72, 1989.
22. B. Kalyanasundaram and K. Pruhs. Online weighted matching. *Journal of Algorithms*, 14(3):478–488, 1993.
23. R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for online bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 352–358. ACM, 1990.
24. I. Katriel, C. Kenyon-Mathieu, and E. Upfal. Commitment under uncertainty: Two-stage stochastic matching problems. *Theoretical Computer Science*, 408(2-3):213–223, 2008.
25. M. Mahdian, H. Nazerzadeh, and A. Saberi. Allocating online advertisement space with unreliable estimates. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 288–294, 2007.
26. V. H. Manshadi, S. O. Gharan, and A. Saberi. Online stochastic matching: Online actions based on offline statistics. In *Proceedings of the 22th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2011.
27. A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. Adwords and generalized on-line matching. In *Proceedings of the 46th Annual Symposium on Foundations of Computer Science*, pages 264–273, 2005.
28. A. Schrijver. *Combinatorial Optimization*. Springer-Verlag, 2003.
29. H. Shachnai and A. Srinivasan. Finding large independent sets in graphs and hypergraphs. *SIAM Journal on Discrete Mathematics*, 18(3):488–500, 2005.
30. D. Shmoys and C. Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *Journal of the ACM*, 53(6):978–1012, 2006.
31. C. Swamy and D. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *ACM SIGACT News*, 37(1):46, 2006.

A Cardinality Constrained Multiple Round Stochastic Matching

We now consider the cardinality constrained multiple round stochastic matching; this was also defined in [9]. In this problem, we arrange for at most C disjoint pairs to date each other simultaneously (constrained by the fact that each person is involved in at most one date at any time), and have k days in which all these dates must happen—again, we want to maximize the expected weight of the matched pairs.

More formally, we can probe several edges concurrently—a "round" may involve probing any set of edges that forms a matching of size at most C. Given k and C, the goal is to find an adaptive strategy for probing edges in rounds such that we use at most k rounds, and maximize the expected weight of matched edges during these k rounds. As before, one can probe edges involving individual i at most t_i times, and only if i is not already matched by the algorithm. In this section, we give a constant-factor approximation for this problem, improving over the previously known $O(\min\{k, C\})$-approximation [9] (which only works for the unweighted case).

Our approach, as in the previous sections, is based on linear programming. Using the same argument in Claim 1, we can show that the optimal value of the following LP is an upper bound on any (adaptive) algorithm for the multiple round stochastic matching problem. Below, $\mathcal{M}_C(G)$ denotes the convex hull of all matchings in G having size at most C.

$$\text{maximize} \sum_{(i,j)\in E} w_{ij} \cdot \sum_{h=1}^{k} x_{ij}^h \tag{LP4}$$

subject to

$$\sum_{h=1}^{k} y_{ij}^h \le 1 \qquad \forall (i,j) \in E \tag{28}$$

$$\sum_{j\in\partial(i)} \sum_{h=1}^{k} y_{ij}^h \le t_i \qquad \forall i \in V \tag{29}$$

$$y^h \in \mathcal{M}_C(G) \qquad \forall h \in [k] \tag{30}$$

$$x_{ij}^h = p_{ij} \cdot y_{ij}^h \qquad \forall (i,j) \in E,\ h \in [k] \tag{31}$$

$$\sum_{j \in \partial(i)} \sum_{h=1}^{k} x_{ij}^h \leq 1 \qquad \forall i \in V \tag{32}$$

Since there is a linear description for $\mathcal{M}_C(G)$, for which we can separate in polynomial time [28, Corollary 18.10a]), the above LP can be solved in polynomial time using the Ellipsoid algorithm. To see that this LP is indeed a relaxation of the original adaptive problem, observe that setting y_{ij}^h to be "probability that (i,j) is probed in round h by the optimal strategy" defines a feasible solution to the LP with objective equal to the optimal value of the stochastic matching instance.

Our algorithm first solves the LP to optimality and obtains solution (x, y). Note that for each $h \in [k]$, using the fact that polytope $\mathcal{M}_C(G)$ is integral and that the variables $y^h \in \mathcal{M}_C(G)$, we can write y^h as a convex combination of matchings of size at most C; i.e., we can find matchings $\{M_\ell^h\}_\ell$ and positive values $\{\lambda_\ell^h\}_\ell$ such that each M_ℓ^h is a matching in G of size at most C and $y^h = \sum \lambda_\ell^h \cdot \chi(M_\ell^h)$, where $\chi(M_\ell^h)$ denotes the characteristic vector corresponding to the edges that the are present in the matching. (See, e.g. [8], for a polynomial-time procedure.) Fixing the parameter α to a suitable value to be specified later, the algorithm does the following.

1. for each round $h = 1, \cdots, k$ do
   a. define the h^{th} matching
   $$\mathbb{P}^h := \begin{cases} \emptyset & \text{with probability } 1 - \frac{1}{\alpha} \\ M_\ell^h & \text{with probability } \frac{\lambda_\ell^h}{\alpha} \end{cases}$$
   b. Probe all edges in $\mathbb{P}^h$ that are safe.

We show that this algorithm is a 20-approximation for $\alpha = 10$, which proves Theorem 6.

As before, an edge $(i,j) \in E$ is said to be *safe* iff (a) (i,j) has not been probed earlier, (b) neither i nor j is matched, and (c) neither i nor j has timed out.

Lemma 12 *For any edge $(i,j) \in E$, and at round $h \in [k]$, $\Pr[\,(i,j)$ is safe in round $h\,] \geq 1 - \frac{5}{\alpha}$.*

Proof: We will show that the following three statements hold at round h:
i. $\Pr[\,(i,j)$ has probed$\,] \leq \frac{1}{\alpha}$.
ii. $\Pr[\,$vertex i is already timed out$\,] \leq \frac{1}{\alpha}$.
iii. $\Pr[\,$vertex i is already matched$\,] \leq \frac{1}{\alpha}$.
Since $\Pr[\,(i,j)$ is *not* safe in round $h\,]$ is at most

$$\Pr[\,(i,j) \text{ been probed}\,] + \Pr[\,i \text{ matched}\,] + \Pr[\,i \text{ timed out}\,] + \Pr[\,j \text{ matched}\,] + \Pr[\,j \text{ timed out}\,]$$

by the trivial union bound, proving (i)-(iii) will prove the lemma. To prove (i), observe that for any edge $e \in E$ and round g, $\Pr[\,e$ probed in round $g\,] \leq \Pr[e \in \mathbb{P}^g\,] = \frac{1}{\alpha} y_e^g$, and hence $\Pr[\,(i,j)$ probed before round $h\,] \leq \frac{1}{\alpha} \sum_{g<h} y_e^g \leq \frac{1}{\alpha}$, where the last inequality uses LP constraint (28).

The proof for (iii) is identical, using the LP constraint (32). Statement (ii) follows from Markov inequality, noting that the expected number of probes on vertex i is at most $\frac{t_i}{\alpha}$. ■

Theorem 10 *Setting $\alpha = 10$ gives a 20-approximation for multiple round stochastic matching.*

Proof: Using Lemma 12, we have for any edge $(i,j) \in E$ and round $h \in [k]$,

$$\begin{aligned} \Pr[\,(i,j) \text{ probed in round } h\,] &= \Pr[\,(i,j) \text{ safe in round } h\,] \cdot \Pr[\,(i,j) \in \mathbb{P}^h \mid (i,j) \text{ safe in round } h\,] \\ &\geq \left(1 - \frac{5}{\alpha}\right) \cdot \Pr[\,(i,j) \in \mathbb{P}^h \mid (i,j) \text{ safe in round } h\,] \\ &= \left(1 - \frac{5}{\alpha}\right) \cdot \frac{y_{ij}^h}{\alpha}, \end{aligned}$$

where the equality follows from the fact that events $(i,j) \in \mathbb{P}^h$ and (i,j) is safe in round h are independent. Thus the expected value accrued by the algorithm is

$$\sum_{e \in E} w_e \cdot \sum_{h=1}^{k} \Pr[\,e \text{ probed in round } h\,] \cdot p_e \geq \frac{1}{\alpha}\left(1 - \frac{5}{\alpha}\right) \cdot \sum_{e \in E} w_e \cdot \sum_{h=1}^{k} y_e^h \cdot p_e,$$

which is $\frac{1}{\alpha}\left(1 - \frac{5}{\alpha}\right)$ times the optimal LP-value. Setting $\alpha = 10$ completes the proof. ■

B Unweighted Stochastic Matching: A Greedy Algorithm

In this section we consider a greedy algorithm for the *unweighted* stochastic matching problem: in this unweighted version, all edges have unit weight, and the goal is to maximize the expected number of matched edges. The greedy algorithm was proposed by Chen et al. [9], and they gave an analysis proving it to be a 4-approximation; however, the proof was fairly involved. Here, we give a significantly simpler analysis showing an approximation guarantee of 5. The greedy algorithm we consider is the following:

1. Let σ denote the ordering of the edges in E by non-increasing p_e-values.
2. Consider the edges $e \in E$ in the order given by σ
   a. If edge e is *safe* then probe it, else do not probe e.

Recall that an edge is safe if neither of its endpoints have been matched or timed out. Note that the expected value of the greedy algorithm is

$$\mathsf{ALG} = \sum_{e \in E} \Pr[\, e \text{ is matched}\,] = \sum_{e \in E} \Pr[\, e \text{ is probed}\,] \cdot p_e.$$

B.1 The Analysis

While the algorithm does not have anything to do with the linear programming relaxation we presented in the previous section, we will use that LP for our analysis. Consider the optimal LP solution (x^*, y^*), and recall that (x^*, y^*) satisfy the conditions (4)-(7). (Alternatively, use the fractional solution $y^*_e := \Pr[\, e \text{ is probed in the optimal strategy}\,]$ and $x^*_e := \Pr[\, e \text{ is matched in the optimal strategy}\,]$.) For each $e = (i, j) \in E$, define the following three events:

$$\begin{aligned} M_e &:= \text{ either } i \text{ or } j \text{ is matched when } e \text{ is considered in } \sigma, \\ R_e &:= \text{ either } i \text{ or } j \text{ has timed out when } e \text{ is considered in } \sigma, \text{ and} \\ B_e &:= M_e \vee R_e. \end{aligned}$$

By the algorithm, it follows that $\Pr[\, e \text{ is probed}\,] = 1 - \Pr[B_e]$ for all $e \in E$. So,

$$\mathsf{ALG} = \sum_{e \in E} (1 - \Pr[B_e])\, p_e \geq \sum_{e \in E} (1 - \Pr[B_e]) \cdot y^*_e\, p_e \tag{33}$$

The following two lemmas charge the value accrued by the algorithm in two different ways to the optimal LP solution.

Lemma 13 $2\mathsf{ALG} \geq \sum_{g \in E} \Pr[M_g] \cdot y^*_g \cdot p_g$.

Proof: In the greedy algorithm, whenever edge $e = (i, j)$ gets matched, *write* value of $\frac{y^*_f \cdot p_f}{2}$ on each edge $f \in \partial(i) \bigcup \partial(j)$. Note that the total value written when edge $e = (i, j)$ gets matched is at most:

$$\sum_{f \in \partial(i)} \frac{y^*_f\, p_f}{2} + \sum_{f \in \partial(j)} \frac{y^*_f\, p_f}{2} = \frac{1}{2} \sum_{f \in \partial(i)} x^*_f + \frac{1}{2} \sum_{f \in \partial(j)} x^*_f \leq 1,$$

where the inequality follows from (4). Recall that in any possible execution of Greedy, an edge is matched at most once. Thus the expected total value written (on all edges) is at most $\sum_{e \in E} \Pr[\, e \text{ is matched}\,] = \mathsf{ALG}$.

On the other hand, whenever event M_g occurs in the greedy algorithm (at some edge $g = (a, b) \in E$), *read* $\frac{y^*_g \cdot p_g}{2}$ value from g. Consider any outcome where event M_g occurs: it must be that either a or b was already matched (say via edge e); this in turn means that $\frac{y^*_g \cdot p_g}{2}$ value was written on edge g at the time when e got matched (since g is adjacent to e). Thus the value read from an edge (at any point) is at most the value already written on it. Thus the expected total value read from all edges is $\sum_{g \in E} \Pr[M_g] \cdot \frac{y^*_g\, p_g}{2} \leq \mathbb{E}[\text{total value written}] \leq \mathsf{ALG}$. ■

Lemma 14 $2\mathsf{ALG} \geq \sum_{g \in E} \Pr[R_g] \cdot y^*_g \cdot p_g$.

Proof: Consider the execution of the greedy algorithm, with a *value* α_e defined on each edge $e \in E$ (initialized to zero). Whenever an edge $e = (i, j)$ gets probed, do (where σ_e denotes the edges in E that appear after e in σ):

1. For each $f \in \partial(i) \cap \sigma_e$, increase α_f by $\frac{y^*_f\, p_f}{2 t_i}$.
2. For each $f \in \partial(j) \cap \sigma_e$, increase α_f by $\frac{y^*_f\, p_f}{2 t_j}$.

Let $A := \sum_{e \in E} \alpha_e$. Note that the increase in A when edge $e = (i, j)$ gets probed is:

$$\sum_{f \in \partial(i) \cap \sigma_e} \frac{y^*_f\, p_f}{2 t_i} + \sum_{f \in \partial(j) \cap \sigma_e} \frac{y^*_f\, p_f}{2 t_j} \leq \frac{p_e}{2} \left(\frac{1}{t_i} \sum_{f \in \partial(i) \cap \sigma_e} y^*_f + \frac{1}{t_j} \sum_{f \in \partial(j) \cap \sigma_e} y^*_f \right) \leq p_e,$$

where for the first inequality we use the greedy property that $p_e \geq p_f$ for all $f \in \sigma_e$ and the second inequality follows from (5). Thus the expected value of A at the end of the greedy algorithm is $\mathbb{E}[A \text{ at the end of Greedy}] \leq \sum_{e \in E} \Pr[\, e \text{ is probed}\,] \cdot p_e = \mathsf{ALG}$. (Recall that in any possible execution of Greedy, an edge is probed at most once.)

On the other hand, whenever event R_g occurs in the greedy algorithm (at some edge $g = (a, b) \in E$), *read* the value α_g from g. Consider any outcome where event R_g occurs: it must be that either a or b was already timed out (say vertex a). This means that t_a edges from $\partial(a)$ have already been probed. By the updates to α-values defined above, since g is adjacent to each edge in $\partial(a)$, the current value $\alpha_g \geq t_a \cdot \frac{y^*_g\, p_g}{2\, t_a} = y^*_g\, p_g/2$. So whenever R_g occurs, the value read $\alpha_g \geq y^*_g\, p_g/2$. I.e. the expected total value read is at least $\sum_{g \in E} \Pr[R_g] \cdot \frac{y^*_g\, p_g}{2}$. However, the total value read is at most the value A at the end of the greedy algorithm. This implies that $\sum_{g \in E} \Pr[R_g] \cdot \frac{y^*_g\, p_g}{2} \leq \mathbb{E}[\text{total value read}] \leq \mathbb{E}[A \text{ at the end of Greedy}] \leq \mathsf{ALG}$. ■

Proof of Theorem 5: Adding the expressions from Lemmas 13 and 14, we get

$$4\,\mathsf{ALG} \geq \sum_{e \in E} (\Pr[M_e] + \Pr[R_e]) \cdot y_e^*\, p_e \geq \sum_{e \in E} \Pr[B_e] \cdot y_e^*\, p_e,$$

where the second inequality uses the definition $B_e = M_e \vee R_e$. Adding this to (33), we obtain $5\,\mathsf{ALG} \geq \sum_{e \in E} y_e^* \cdot p_e$, which is the optimal LP objective. Thus, the greedy algorithm is a 5-approximation. ■