Bipartite Stochastic Matching: Online, Random Order, and I.I.D. Models

Allan Borodin [*] Calum MacRury [†] Akash Rakheja [‡]

Abstract

Within the context of stochastic probing with commitment, we consider the online stochastic matching problem; that is, the one sided online bipartite matching problem where edges adjacent to an online node must be probed to determine if they exist, based on known edge probabilities. If a probed edge exists, it must be used in the matching (if possible). We study this problem in the generality of a patience (or timeout) constraint which limits the number of probes that can be made to edges adjacent to an online node. Arbitrary patience constraints result in modelling and computational efficiency issues that are not encountered in the special cases of unit patience and full (i.e., unlimited) patience. The stochastic matching problem leads to a variety of settings. Our main contribution is to provide a new LP relaxation and a unified approach for establishing new and improved competitive bounds in three different input model settings (namely, adversarial, random order, and known i.i.d.). In all these settings, the algorithm does not have any control on the ordering of the online nodes. We establish competitive bounds in these settings, all of which generalize the standard non-stochastic setting when edges do not need to be probed (i.e., exist with certainty). All of our results hold for arbitrary edge probabilities and patience constraints. Specifically, we establish the following competitive ratio results:

1. A $1-1/e$ ratio when the stochastic graph is known, offline vertices are weighted and online arrivals are adversarial.
2. A $1-1/e$ ratio when the stochastic graph is known, edges are weighted, and online arrivals are given in random order (i.e., in ROM, the random order model).
3. A $1-1/e$ ratio when online arrivals are drawn i.i.d. from a known stochastic type graph and edges are weighted.
4. A (tight) $1/e$ ratio when the stochastic graph is unknown, edges are weighted and online arrivals are given in random order.

We note that while results for stochastic graphs in the ROM setting generalize the corresponding results for the classical ROM bipartite matching setting, it is not clear that a result for a known stochastic graph in the ROM setting implies the same result for the stochastic unknown and known i.i.d. settings.

In deriving our results, we clarify and expand upon previous offline benchmarks, relative to which one defines an appropriate definition of the competitive ratio. In particular, we introduce a new LP relaxation which upper bounds the performance of "an ideal benchmark".

[*] Department of Computer Science, University of Toronto, Toronto, ON, Canada bor@cs.toronto.edu

[†] Department of Computer Science, University of Toronto, Toronto, ON, Canada cmacrury@cs.toronto.edu

[‡] Department of Computer Science, University of Toronto, Toronto, ON, Canada rakhejaakash@gmail.com

Contents

1 Introduction 3

2 Preliminaries and Techniques 4

2.1 A Review of Our Technical Contributions . . . 5

2.2 Defining a New LP . . . 8

2.3 An Overview of Our Results . . . 11

3 Known Stochastic Graphs: Adversarial and ROM Arrivals 13

3.1 Defining the Probing Algorithm . . . 13

3.2 Adversarial Arrivals . . . 13

3.3 Random Order Arrivals . . . 16

3.3.1 Improving Upon the Competitive Ratio . . . 17

4 Online Stochastic Matching in the Known I.I.D. Model 21

4.1 The Known I.I.D. Stochastic Setting . . . 22

4.2 Defining an LP Relaxation . . . 23

4.3 Defining a Known I.I.D. Probing Algorithm . . . 24

5 Online Stochastic Matching with ROM Arrivals: The Case of an Unknown Stochastic Graph 27

6 Conclusion and Open Problems 32

A Relaxing the Committal Benchmark 35

A.1 Non-adaptivity in the Relaxed Stochastic Matching Problem . . . 38

B Solving LP-new Efficiently 42

C The Non-committal Benchmark 43

C.1 Relaxing the Non-committal Benchmark via LP-std . . . 44

C.2 LP-new and the Non-committal Benchmark . . . 46

D LP Relations 48

E Non-adaptive Probing Algorithms and Adaptivity Gaps 51

F Deferred Proofs 51

G Extended Related Works 55

1 Introduction

Stochastic probing problems are part of the larger area of decision making under uncertainty and more specifically, stochastic optimization. Unlike more standard forms of stochastic optimization, it is not just that there is some stochastic uncertainty in the set of inputs, stochastic probing problems involve inputs that cannot be determined without probing (at some cost and/or within some constraint). Applications of stochastic probing occur naturally in many settings, such as in matching problems where compatibility cannot be determined without some trial or investigation (for example, in online dating and kidney exchange applications). There is by now an extensive literature for stochastic matching problems. For space efficiency, we will give an extended overview of related work in Appendix G. Research most directly relating to this paper will appear as we proceed.

The *stochastic matching problem*[1] was introduced by Chen et al. [12]. In this problem, we are given an adversarially generated *stochastic graph* $G = (V, E)$ with a probability p_e associated with each edge e and a patience (or timeout) parameter ℓ_v associated with each vertex v. An algorithm probes edges in E within the constraint that at most ℓ_v edges are probed incident to any particular vertex $v \in V$. The patience constraint can be viewed as a simple budgetary constraint, where each probe has unit cost and the patience constraint is the budget. When an edge e is probed, it is guaranteed to exist with probability exactly p_e. If an edge (u, v) is found to exist, it is added to the matching and then u and v are no longer available. The goal is to maximize the expected size of a matching constructed in this way. This problem can be generalized to offline vertices or edges having weights and then the objective is to maximize the expected weight of the matching. Notably, in Chen et al., the algorithm knows the entire stochastic graph in advance.

In addition to generalizing the setting of the results of Chen et al., Bansal et al. [6] introduced an i.i.d. bipartite version of the problem where nodes on one side of the partition arrive online and edges adjacent to that node are then probed. In their model, each online vertex (and its adjacent edges) is drawn independently and identically from a known distribution. That is, the possible "type" of each online node (i.e., the adjacent edge probabilities and edge weights) is known and the input sequence is then determined i.i.d. from this known distribution, where the type of a node is presented to the algorithm upon arrival. In both the Chen et al. and Bansal et al. models, each offline node has unlimited patience, whereas each online node specifies its patience upon arrival. As in other online bipartite matching problems, the match for an online node must be made before the next online arrival. In both of these models, if an edge is probed and confirmed to exist, then it must be included in the current matching (if possible). This problem is referred to as the *online stochastic matching problem*[2] (with patience) and also referred to as the *stochastic rewards problem*, though we avoid the latter terminology. In various settings, we will study the online stochastic matching problem. More specifically, we will consider online settings where the algorithm knows the adversarially determined stochastic graph, where a stochastic (type) graph and a distribution on the online vertices is known and online nodes are generated i.i.d. from this distribution, and in the random order model (ROM) when the stochastic graph is both known and unknown[3]. Amongst other applications, the online stochastic matching problem notably models

[1] Unfortunately, the term "stochastic matching" is also used to refer to more standard optimization where the input (i.e., edges or vertices) are drawn from some known or unknown distributions but no probing is involved.

[2] The online stochastic matching problem is sometimes meant to imply unit patience but we will mainly be interested in arbitrary patience values.

[3] In a related paper, we establish a $1 - 1/e$ competitive ratio in the setting where the stochastic graph is unknown and the offline vertices are weighted. This setting is simpler and allows for a completely combinatorial deterministic algorithm. It is interesting to note that in this setting the same algorithm can be derived using the LP based approach of this paper but the combinatorial method is conceptually and computationally simpler [8].

online advertising where the probability of an edge can correspond to the probability of a purchase in online stores or to pay per click revenue in online searching.

We note that these stochastic matching models generalize the corresponding classical non-stochastic models where edges adjacent to an online node are known upon arrival and do not need to be probed. It follows that any inapproximation results in the classical setting apply to the corresponding stochastic setting.

2 Preliminaries and Techniques

The online stochastic matching problem generalizes the classical online bipartite setting as follows. For each $e \in E$ in the stochastic (bipartite) graph $G = (U, V, E)$, there is a fraction $0 \leq p_e \leq 1$ associated with e that gives the probability of existence of the edge e. More precisely, each edge $e \in E$ is associated with an independent Bernoulli random variable of parameter p_e, which we denote by $\mathrm{st}(e)$, corresponding to the **state** of the edge. If $\mathrm{st}(e) = 1$, then we say that e is **active**, and otherwise we say that e is **inactive**. It will be convenient to hereby assume that $E = U \times V$. In this way, if we wish to exclude a pair $(u, v) \in U \times V$ from existing as an edge in G, then we may set $p_{u,v} = 0$, thus ensuring that (u, v) is always inactive.

When an online node $v \in V$ arrives, the **online probing algorithm** sees all the adjacent edges and associated probabilities but must perform a **probing operation** on the edge to reveal/expose its state, $\mathrm{st}(e)$. As in the classical problem, an online algorithm must decide on a possible match for an online node v before seeing the next online node. The algorithm can be **non-greedy** and not match a given $v \in V$ even though some $u \in U$ is still unmatched. The online stochastic matching problem simplifies to the classical setting in one of two ways: (1) if $p_e \in \{0, 1\}$ for all edges e, or (2) if the algorithm is allowed to probe all edges adjacent to an online node v before determining which, if any, node to match to v. To make stochastic probing problems meaningful, we either must have a cost for probing or some kind of **commitment** upon probing an input item. Specifically, if an edge $e = (u, v)$ is probed in the stochastic matching problem and turns out to be active, then e must be added to the current matching, provided u and v are both currently unmatched. We say that the online probing algorithm **respects commitment** or is **commital**, provided it satisfies this property. Furthermore, in the stochastic matching problem, for each online node v, there is a known **patience parameter** (also called timeout) ℓ_v that bounds the number of probes that can be made to edges adjacent to v. The classical online bipartite matching problems (unweighted, vertex weighted, or edge weighted) for adversarial, ROM, and i.i.d. input sequences all generalize to the stochastic matching setting. We emphasize that the online stochastic matching problem generalizes the classical online problem, even when restricted to the case of **unit patience** (i.e., $\ell_v = 1$ for all $v \in V$).

Clearly, in the classical adversarial or ROM settings, if the algorithm knew the input graph G, the online algorithm could compute an optimal solution before seeing the online sequence and use that optimal solution to determine an optimal matching online. But similar to knowing the type graph in the classical setting with i.i.d. inputs, an algorithm still lacks the ability to know the states of the edges of G, namely $(\mathrm{st}(e))_{e \in E}$, so that the stochastic matching problem is interesting, whether the stochastic graph G is known or unknown to the algorithm. We are left then with a wide selection of problems, depending on whether or not the stochastic graph is known, how input sequences are determined, and whether or not edges or vertices are weighted. In the classical i.i.d. bipartite matching problem, competitive bounds for an unknown distribution follow from the corresponding ROM problem by a result of Karande et al. [27]. The same result (using the same argument) holds in the stochastic matching setting, provided the stochastic graph is unknown. It is unclear to us whether this reduction continues to hold for the ROM setting when the stochastic graph is known, as we expand upon in Section 4. We will focus on the following settings: a known stochastic graph with adversarial and ROM inputs, the i.i.d. setting with a known stochastic type graph and distribution on the online vertices, and an unknown stochastic graph with ROM arrivals and weighted edges.

What is the benchmark against which we measure the competitive performance of an online algorithm in the stochastic matching problem? In the classical online setting, we compare the value of the online algorithm to that of an optimal matching of the graph. If the inputs are drawn from a distribution, we then compare the expected value of the algorithm to the expected value of an optimum matching. For stochastic probing problems, it is easy to see we cannot hope to obtain a reasonable competitive bound for this type of comparison; that is, if we are comparing the expected value of an online probing algorithm to the expected value of an optimum matching of the stochastic graph. For example, consider a single online vertex with patience 1, and n offline (unweighted) vertices where each edge e has probability $\frac{1}{n}$ of being present. The expectation of an online probing algorithm will be at most $\frac{1}{n}$ while the expected size of an optimal matching (over all instantiations of the edge probabilities) will be $1-(1-\frac{1}{n})^n \rightarrow 1-\frac{1}{e}$. This example clearly shows that no constant ratio is possible if the patience is sublinear (in $n=|U|$).

A reasonable approach is to force the benchmark to adhere to the commitment and patience requirements of G that the online algorithm satisfies. Following previous work and the explicit reasoning in Brubach et al.[9], an ideal benchmark is the following: knowing the stochastic graph G (or G and the type graph in the stochastic i.i.d. setting) and the patience requirements of the online nodes, the benchmark can probe edges in any *adaptive* order but must satisfy the commitment and patience requirements of the online vertices. By adaptive order, we mean that the next edge to be probed will depend on all the edges that have been currently revealed and the current matching. We emphasize that this benchmark is not restricted to any ordering of the online vertices. In particular, we note that after probing some edge (u_1, v_1), the next probed edge can be (u_2, v_2) where u_2 and v_2 each may be distinct from u_1 and v_1, respectively. As in online probing algorithms, the goal of the benchmark is to build a matching whose weight is as large as possible in expectation. We refer to this benchmark as the **committal benchmark**, and denote the expected value of its matching by $\mathrm{OPT}(G)$.

We also consider a *stronger* benchmark which still must adaptively probe edges subject to patience constraints, but isn't restricted by commitment; that is, it may decide upon which subset of edges to match *after* all its probes have been made. Once again, the probes of this benchmark need not respect any ordering on the online nodes, and the benchmark's goal is to build a matching of maximum expected weight. We refer to this benchmark as the **non-committal benchmark**, and denote the expected value of the matching it constructs by $\mathrm{OPT}_{non}(G)$. Observe that in the case of **full patience** (i.e., $\ell_v = |U|$ for all $v \in V$), the benchmark may probe all the edges of G, and thus corresponds to the expected weight of the optimum matching of the stochastic graph.

Following what is standard in the stochastic matching literature, we prove our results against the committal benchmark, though in Appendix C we prove that our results also hold against the non-committal benchmark for the case of unit or full patience values.

2.1 A Review of Our Technical Contributions

Suppose we are presented a stochastic bipartite graph $G=(U,V,E)$, with edge probabilities $(p_e)_{e \in E}$, edge weights $(w_e)_{e \in E}$ and patience values $(\ell_v)_{v \in V}$. Here V is the set of online vertices and U is the set of offline vertices. We assume U is known apriori to an online algorithm and the vertices in V arrive online. In both the committal and non-committal offline benchmarks, it is not clear how to compute these optimal benchmark values. As such, one instead resorts to an appropriate LP upper bound on their value.

The most prevalent (standard) LP used in the literature was introduced by Bansal et al. [6], where each pair (u, v) for $u \in U$ and $v \in V$ has a variable $x_{u,v}$, corresponding to the probability that the committal benchmark probes (u, v).

$$\begin{aligned}
&\text{maximize} && \sum_{u \in U, v \in V} w_{u,v} \cdot p_{u,v} \cdot x_{u,v} && && \text{(LP-std)}\\
&\text{subject to} && \sum_{v \in V} p_{u,v} \cdot x_{u,v} \leq 1 && \forall u \in U && (2.1)\\
& && \sum_{u \in U} p_{u,v} \cdot x_{u,v} \leq 1 && \forall v \in V && (2.2)\\
& && \sum_{u \in U} x_{u,v} \leq \ell_v && \forall v \in V && (2.3)\\
& && 0 \leq x_{u,v} \leq 1 && \forall u \in U, v \in V. && (2.4)
\end{aligned}$$

Bansal et al. [6] observed that if $\text{LPOPT}_{std}(G)$ denotes the value of an optimal solution to LP-std, then it is a **relaxation** of the committal benchmark OPT(G); that is,

$$\text{OPT}(G) \leq \text{LPOPT}_{std}(G). \tag{2.5}$$

To date, the most common technique for proving guarantees against the committal benchmark involves comparing the performance of one's probing algorithm to $\text{LPOPT}_{std}(G)$, as opposed to $\text{OPT}(G)$ directly. In fact, when G is *known* to the algorithm, one can leverage a solution to LP-std to determine which probes one should make (see [6, 3, 10, 11, 7]).

This is especially effective in the case of unit patience, as an optimum solution to LP-std, say $(x_{u,v})_{u \in U, v \in V}$, induces a distribution for each vertex $v \in V$. Rather,

$$\sum_{u \in U} x_{u,v} \leq 1,$$

for each $v \in V$. Thus, when processing an arriving online node v, one can choose to probe $u \in U$ with probability $x_{u,v}$ (where one *passes* on v with probability $1 - \sum_{u \in U} x_{u,v}$). While this ignores the issue of the online nodes *colliding* (i.e., multiple vertices of V attempting a match to $u \in U$), (2.1) ensures each vertex of U is matched at most once *in expectation*. In the case of offline vertex weights, this is sufficient to prove a guarantee of $1 - 1/e$ against $\text{LPOPT}_{std}(G)$, no matter how the vertices of V are presented to the probing algorithm. Similarly, in the case of edge weights, this approach suffices to achieve a guarantee of $1/2$, albeit requiring the vertices of V to arrive in random order. We emphasize that G must be known to the probing algorithm in order to implement these strategies. Both these arguments (in the more general setting of arbitrary patience) are discussed in detail in Section 3.

If one now moves to the case when v has arbitrary patience ℓ_v, then the values $(x_{u,v})_{u \in U}$ satisfy the following inequalities:

$$\sum_{u \in U} x_{u,v} \leq \ell_v \quad \text{and} \quad \sum_{u \in U} p_{u,v} \cdot x_{u,v} \leq 1. \tag{2.6}$$

For a fixed v, the techniques in [6, 3, 10, 7, 11] involve first drawing a random ordering π of U (where different distributions are used in different papers). Once this is done, a random subset $P \subseteq U$

is drawn using the GKSP algorithm of Gandhi et al. [22], with the guarantee that $u \in P$ with probability $x_{u,v}$, and that $|P| \leq \ell_v$. The edges $(u_i, v)_{i=1}^{|P|}$ are then probed in the order $u_1, \ldots, u_{|P|}$ induced by π. Since the algorithm must respect commitment, (u_i, v) is probed with probability $\prod_{j=1}^{i-1}(1 - p_{u_j,v})$.

Clearly this GKSP rounding approach is problematic, as a fixed vertex $u \in U$ may get probed with probability sufficiently less than $x_{u,v}$, depending on how highly prioritized it is in the ordering π. The solutions in the literature involve drawing π in such a way that each vertex $u \in U$ is probed with probability as *close* to $x_{u,v}$ as possible, but it is clear that no approach is without loss for *every* vertex of U.

While this describes the major challenge with generalizing to arbitrary patience, in theory it does not preclude a probing algorithm from existing which matches the ratio of $1 - 1/e$, as attainable in the vertex weighted unit patience case. Unfortunately, the ratio between $\text{OPT}(G)$ and $\text{LPOPT}_{std}(G)$ can become quite small, depending on the values of $(\ell_v)_{v \in V}$ and the instance G. In [9], Brubach et al. define the **stochasticity gap** of this LP as the infimum of this ratio across all stochastic graphs, namely $\inf_G \text{OPT}(G)/\text{LPOPT}_{std}(G)$. Clearly, the notion of a stochasticity gap can be extended to any LP relaxation of the committal benchmark, as we shall later do in the context of our new LP.

Brubach et al. also consider the following example, thus providing a negative result for the stochasticity gap of LP-std.

Example 2.1 ([9]). *Fix $n \geq 1$, and construct an unweighted graph $G_n = (U, V, E)$. Suppose that $|U| = |V| = n$ and $\ell_v = n$ for all $v \in V$. Set $E := U \times V$, and define $p_{u,v} := 1/n$ for each $(u, v) \in E$. Observe that G_n corresponds to the Erdős–Rényi random graph $\mathbb{G}_{n,n,1/n}$. In this case,*

$$\mathbb{E}[OPT(G_n)] \leq 0.544 \cdot (1 + o(1))\, LPOPT_{std}(G_n),$$

where the asymptotics are over $n \to \infty$ [4].

Thus, any probing algorithm which attains a guarantee against $\text{LPOPT}_{std}(G)$ has a provable competitive ratio of at most 0.544.

In order to get around this limitation, Gamlath et al. [21] consider an LP in the setting of full patience, which imposes exponentially many constraints, in addition to those of LP-std. Specifically, for each $v \in V$ and $S \subseteq U$, they ensure that

$$\sum_{u \in S} p_{u,v} \cdot x_{u,v} \leq 1 - \prod_{u \in S}(1 - p_{u,v}). \tag{2.7}$$

Observe that in the variable interpretation of LP-std, the left-hand side corresponds to the probability a probing algorithm makes a match to a vertex of S, and the right-hand side corresponds to the probability an edge between v and S exists[5]. The goal of these additional constraints is thus to force the LP to better capture the behavior of the committal benchmark[6].

Using a polynomial time oracle, Gamlath et al. argue that their LP remains poly-time solvable, despite having exponentially many constraints. As in the setting LP-std, they solve their LP to

[4] The example in Brubach et al [9] can clearly be extended to the case when G_n has linearly sized patience, that is when $\min_{v \in V} \ell_v = \Omega(n)$, at the expense of the strength of their negative result (the constant 0.544).

[5] The LP considered by Gamlath et al. in [21] also places the analogous constraints of (2.7) on the vertices of U. That being said, these additional constraints are not used anywhere in the work of Gamlath et al., so we do not consider them when we provide a generalization of their LP in A

[6] In fact, the results of Gamlath et al. are proven against the optimum expected matching of G, which is equivalent to the non-committal benchmark, as they work exclusively in the setting of full patience.

attain a solution $(x_{u,v})_{u \in U, v \in V}$ for G. Each time an online vertex $v \in V$ then arrives, they draw a random permutation π_v on a random subset of U, which indicates both the probes they intend to make, and the order they intend to make them in. By following π_v, their procedure is *lossless*; that is, they are able to probe $u \in U$ with probability exactly $x_{u,v}$[7].

Unfortunately, the results of Gamlath et al., as well as related techniques of Costello et. al [13], do not seem to naturally extend to arbitrary patience. For instance, even the correct modification of (2.7) is not clear to us. However, we are able to provide a reasonable extension of the Gamlath et al. LP in Appendix A.

2.2 Defining a New LP

In this section, we design a new LP for the problem of designing a fixed vertex *lossless* probing algorithm, which works no matter the edge probabilities, edge weights and patience values $(\ell_v)_{v \in V}$ of $G = (U, V, E)$. Instead of attempting to find the appropriate constraints on the variables of LP-std, we take a different approach. Specifically, we ensure our LP has polynomially many constraints, while allowing it exponentially many variables to better indicate how the committal benchmark make decisions.

For each $i \geq 1$, denote $U^{(i)}$ as the collection of tuples of length i constructed from U whose entries are all distinct. Moreover, set $U^{(\leq i)} := \cup_{j=1}^{i} U^{(j)}$.

For each $v \in V$, $1 \leq k \leq \ell_v$ and $\boldsymbol{u} \in U^{(k)}$, define

$$g_v^i(\boldsymbol{u}) := p_{u_i,v} \cdot \prod_{j=1}^{i-1} (1 - p_{u_j,v}),$$

where $\boldsymbol{u} = (u_1, \ldots, u_k)$, and $i \in [k]$ (here $[k] := \{1, \ldots, k\}$). Observe that if one reveals the edge states $(\text{st}(u_i, v))_{i=1}^{k}$ in order, then $g_v^i(\boldsymbol{u})$ corresponds to the probability that (u_i, v) is the first active edge revealed.

We also define a variable, denoted $x_v(\boldsymbol{u})$, which may loosely be interpreted as the likelihood the committal benchmark probes the vertices in the order specified by $\boldsymbol{u} = (u_1, \ldots, u_k)$. These definitions lead to the following LP:

$$\text{maximize} \quad \sum_{v \in V} \sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} \left(\sum_{i=1}^{|\boldsymbol{u}|} w_{u_i,v} \, g_v^i(\boldsymbol{u}) \right) \cdot x_v(\boldsymbol{u}) \tag{LP-new}$$

$$\text{subject to} \quad \sum_{v \in V} \sum_{i=1}^{\ell_v} \sum_{\substack{\boldsymbol{u}^* \in U^{(\leq \ell_v)}: \\ u_i^* = u}} g_v^i(\boldsymbol{u}^*) \cdot x_v(\boldsymbol{u}^*) \leq 1 \qquad \forall u \in U \tag{2.8}$$

$$\sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} x_v(\boldsymbol{u}) \leq 1 \qquad \forall v \in V, \tag{2.9}$$

$$x_v(\boldsymbol{u}) \geq 0 \qquad \forall v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)} \tag{2.10}$$

LP-new is a relaxation of the committal benchmark. However, unlike many of the LP formulations in the stochastic matching literature, we are not aware of an immediate proof of either of

[7] The results of Costello et. al [13] also consider the full patience non-bipartite stochastic matching problem, though without edge weights. They derive a probing strategy for a fixed vertex $v \in V$ of $G = (V, E)$ and its neighbourhood $N(v)$, which attains the same guarantee as that of Gamlath et al. [21] through combinatorial techniques.

these facts. We instead must introduce a related stochastic probing problem, known as the **relaxed stochastic matching problem**, which is exactly encoded by LP-new, and whose optimum value upper bounds the committal benchmark. We provide the relevants definitions in Appendix A, where we also prove the following theorem:

Theorem 2.2. *For any stochastic graph G, an optimum solution to LP-new upper bounds $OPT(G)$, the value of the committal benchmark on G.*

Not only is LP-new a relaxation of the committal benchmark, it also can be solved efficiently. To see this, we first take its dual:

$$\begin{aligned}
&\text{minimize} && \sum_{u \in U} \alpha_u + \sum_{v \in V} \beta_v && && \text{(LP-new-dual)}\\
&\text{subject to} && \beta_v + \sum_{j=1}^{|\boldsymbol{u}^*|} g_v^j(\boldsymbol{u}^*) \cdot \alpha_{u_j^*} \geq \sum_{j=1}^{|\boldsymbol{u}^*|} w_{u_j^*,v} \cdot g_v^j(\boldsymbol{u}^*) && \forall v \in V, \boldsymbol{u}^* \in U^{(\leq \ell_v)} && (2.11)\\
& && \alpha_u \geq 0 && \forall u \in U && (2.12)\\
& && \beta_v \geq 0 && \forall v \in V && (2.13)
\end{aligned}$$

In Appendix B, we argue that LP-new-dual has a polynomial time separation oracle, by solving an optimization problem similar to the one considered by Brubach et al. [9]. By standard duality techniques involving the ellipsoid algorithm [35, 23], this allows us to find a solution to LP-new in polynomial time, no matter the patience values of G (see [40, 2, 30] for similar examples). That being said, LP-new clearly always has an optimum solution, which can be found efficiently when $\ell_{max} := \max_{v \in V} \ell_v$ is a constant, independent of the size of $|U|$. Moreover, our results are all in the context of competitive analysis, and so the ratios we present in the various online stochastic matching models all hold, independently of the fact that LP-new can be solved in poly-time.

Suppose now that we are presented a feasible solution, say $(x_v(\boldsymbol{u}))_{\boldsymbol{u} \in U^{(\leq \ell_v)}, v \in V}$, to LP-new, for the stochastic graph $G = (U, V, E)$. For each $v \in V$ and $u \in U$, define

$$\widetilde{x}_{u,v} := \sum_{i=1}^{\ell_v} \sum_{\substack{\boldsymbol{u}^* \in U^{(\leq \ell_v)}: \\ u_i^* = u}} \frac{g^i(\boldsymbol{u}^*) \cdot x_v(\boldsymbol{u}^*)}{p_{u,v}}. \tag{2.14}$$

In order to simplify our notation in the later sections, we refer to the values $(\widetilde{x}_{u,v})_{u \in U, v \in V}$ as the **(induced) edge variables** of the solution $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$.

If we now fix $s \in V$, then we can easily leverage constraint (2.9) to argue that the edge variables $(\widetilde{x}_{u,s})_{u \in U}$ can be probed without loss. Specifically, we may execute the following *fixed vertex* probing algorithm, which we refer to as VertexProbe:

Algorithm 1 VertexProbe

```
Input the stochastic graph $G = (U, V, E)$, a fixed node $s \in V$ and the variables $(x_s(\boldsymbol{u}))_{\boldsymbol{u} \in U^{(\leq \ell_s)}}$
associated to $s$ in a solution to LP-new for $G$.
Initialize $\mathcal{M} \leftarrow \emptyset$.
Return $\mathcal{M}$ with probability $1 - \sum_{\boldsymbol{u} \in U^{(\leq \ell_s)}} x_s(\boldsymbol{u})$        ▷ *pass* with a certain probability..
Draw $\boldsymbol{u}^*$ from $U^{(\leq \ell_s)}$ with probability $x_s(\boldsymbol{u}^*)$ (see (2.9)).
Denote $\boldsymbol{u}^* = (u_1^*, \ldots, u_k^*)$ for $k := |\boldsymbol{u}^*|$.
for $i = 1, \ldots, k$ do
    Probe $(u_i^*, s)$.
    if $\mathrm{st}(u_i^*, s) = 1$ then
        Set $\mathcal{M}(s) \leftarrow u_i^*$ and return $\mathcal{M}$.
    end if
end for
Return $\mathcal{M}$.
```

Observe the following claim, which follows immediately from the definition of the edge variables, $(\widetilde{x}_{u,v})_{u \in U, v \in V}$:

Lemma 2.3. *Let $G = (U, V, E)$ be a stochastic graph with LP-new solution $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$, and whose induced edge variables we denote $(\widetilde{x}_{u,v})_{u \in U, v \in V}$. If the* VertexProbe *algorithm is passed a fixed node $s \in V$, then each node $u \in U$ is probed with probability $\widetilde{x}_{u,s}$.*

Moreover, the edge (u, s) is returned by the algorithm with probability $p_{u,v} \cdot \widetilde{x}_{u,s}$.

Remark. *We say that* VertexProbe ***commits*** *to the edge (u, s), provided the algorithm outputs this edge when executing on the fixed node $s \in V$.*

Before providing an overview of our results, we provide a general algorithmic template which unifies how all of our online probing algorithms are implemented[8].

Let $G = (U, V, E)$ be an adversarially generated stochastic graph, with arbitrary patience values, edge weights and edge probabilities. We may always assume that the online probing algorithm has access to the offline vertices U, but its information regarding V and E is limited, depending on the online model we work in. In general however, we always assume that the online vertices of V are presented in some order to the online probing algorithm, say $v_1, \ldots, v_n$ where $n = |V|$, either through an adversarial, ROM or i.i.d arrival process. We refer to a vertex v_t as **arriving** at **time** $1 \leq t \leq n$. We then follow the general high level template for defining online probing algorithms:

[8] In Section 3, we must use a modified VertexProbe subroutine in the ROM setting with edge weights to improve the competitive ratio from $1/2$ to $1 - 1/e$.

Algorithm 2 General Template

```
   Input $U$ of $G = (U, V, E)$, as well as the remaining information regarding $G$, depending on the online model.
1: $\mathcal{M} \leftarrow \emptyset$.
2: for $t = 1, \ldots, n$ do
3:     Using $U$ and the probing decisions of the previous arrivals (together with $G = (U, V, E)$ if it is known), compute a stochastic graph $H_t = (U_t, V_t, E_t)$ which contains arrival $v_t$, and satisfies $U_t \subseteq U$.
4:     Compute an optimum solution of LP-new for $H_t$, say $(x_v(\boldsymbol{u}))_{v \in V_t, \boldsymbol{u} \in U_t^{(\leq \ell_v)}}$.
5:     Set $e_t \leftarrow \textsc{VertexProbe}(H_t, (x_{v_t}(\boldsymbol{u}))_{\boldsymbol{u} \in U_t^{(\leq \ell_{v_t})}}, v_t)$.
6:     If $e_t \neq \emptyset$, then denote $e_t = (u_t, v_t)$, and if $u_t$ is currently unmatched, set $\mathcal{M}(v_t) \leftarrow u_t$.
7: end for
8: Return $\mathcal{M}$.
```

We once again emphasize that the online probing algorithm has no control over the order of the arrivals, so step (3) is the only place this algorithmic template can be modified. Choosing the "correct" choice of H_t depends on how much information we are privy to (e.g., is the stochastic graph known or unknown), and whether we wish our probing algorithm to execute **adaptively** - that is, depend upon the *probing outcomes* of the previous nodes, $v_1, \ldots, v_{t-1}$ – or not, that is, execute **non-adapatively**[9]. We note that in our results, we will execute Algorithm 2 non-adaptively and the resulting algorithms will be non-greedy. In the subsequent sections, we investigate these issues in detail, and attempt to attain or approach the same competitive ratios one can get in the classical (non-stochastic) online matching settings (when there is a meaningful generalization). All of our results are proven by comparing the performance of the relevant online probing algorithm to LP-new. We are typically able to make use of the classical techniques in the literature (with some key modifications), and so we again emphasize that one of our main technical contributions is in generalizing to arbitrary patience from the more tractable unit/full patience settings and the new LP that is used to derive our results. We also argue that many of the results in the stochastic matching literature actually hold against the non-committal benchmark, as we discuss in detail in Appendix C for the case of general (i.e., not necessarily bipartite) stochastic graphs. We argue this by proving that LP-std is a relaxation of the non-committal benchmark.

2.3 An Overview of Our Results

With these definitions in mind, we now reiterate and point ahead to our main results as first stated in our abstract. All of our results apply to arbitrary patience and the competitive ratios are with respect to the committal benchmark.

1. Theorem 3.1 shows that Algorithm 3 is an online algorithm with competitive ratio $1 - \frac{1}{e}$ in the following stochastic setting:

    - There is a known stochastic graph
    - Online vertices are given adversarially

[9] When processing an online node arrival, say v_t, a **non-adaptive** online probing algorithm can base its probes of v_t on the identities of the previous vertex arrivals, say $v_1, \ldots, v_{t-1}$, as well as their edge weights, edge probabilities and patience values (as well as G itself if it is known). The probes of v_t *cannot* however depend upon the previously probed edge states of $(\text{st}(u, v_k))_{u \in U, k \in [t-1]}$.

- Offline vertices have weights

This result shows that the .544 inapproximation bound against the LP relaxation in Bansal et al. [6] does not hold with respect to our new LP relaxation (see Example 2.1).

2. Theorem 3.8 shows that Algorithm 5 is an online algorithm with competitive ratio $1 - \frac{1}{e}$ in the following setting:

   - There is a known stochastic graph
   - Online vertices are presented in an order determined by a uniform at random permutation of the online vertices in the stochastic graph (i.e., stochastic random order model)
   - Edges have weights

   This algorithm generalizes[10] the online probing algorithm considered by Gamlath et. al [21] in what they refer to as the *query-commit model*.

3. Theorem 4.3 shows that Algorithm 6 is an online algorithm with competitive ratio $1 - \frac{1}{e}$ in the following stochastic i.i.d. setting (improving upon the previously best ratio of 0.46 in [9]):

   - There is a known stochastic (type) graph
   - Online vertices are drawn independently and identically from a distribution on the online vertices (with their adjacent stochastic edges)
   - Edges have weights

   In the classical i.i.d. setting with non-integral arrival rates, Manshadi et al. [32] present an example that shows that $1 - 1/e$ is optimal for **classically non-adaptive**[11] algorithms. Our algorithm fits this classical definition and applies to non-integral arrival rates and hence our algorithm has an optimal competitive ratio amongst this restricted class of probing algorithms.

4. Theorem 5.2 shows that Algorithm 7 is an online algorithm with (tight) competitive ratio $\frac{1}{e}$ in the following setting:

   - The stochastic graph is not known to the algorithm
   - Online vertices are given in random order
   - Edges have weights

   This generalizes the classical non-stochastic result of Kesselheim et al. [29].

All of our probing algorithms are randomized and implemented *non-adaptively*. In Appendix E we discuss the implications of the non-adaptivity by considering the relevant **adaptivity gaps** of the online stochastic matching problems we consider. Roughly speaking, an adaptivity gap is the worst case ratio of performance between the *optimum* non-adaptive probing algorithm, and the committal benchmark[12].

[10] In Appendix C, we provide a reasonable generalization of the Gamlath et al. LP, and show that it attains the same value as LP-new.

[11] Manshadi et al. [32] use the terminology non-adaptive to mean that a (classical) online algorithm in the known i.i.d. setting uses only the *type* of the arriving node to determine its matching decisions. Observe that this restriction is sufficient to ensure that an online probing algorithm is non-adaptive (by our definition) in the stochastic matching setting.

[12] We provide a more precise definition of adaptivity gaps in Appendix E.

3 Known Stochastic Graphs: Adversarial and ROM Arrivals

In this section, we restrict our attention to online bipartite stochastic matching in the setting where the stochastic graph is known to the algorithm. We use our new LP to guide the sequence of probes for each of the online vertices. We first prove Theorem 3.1, showing that Algorithm 3 achieves a $1 - \frac{1}{e}$ competitive ratio for the setting of offline vertex weights and adversarial online arrivals.

We then consider Algorithm 3 in the case of arbitrary edge weights, under the assumption that the online nodes arrive in random order, thus attaining a competitive ratio of $1/2$ (which we show is tight for this algorithm). By considering a modification of Algorithm 3, we can improve this competitive ratio to $1 - 1/e$ using the techniques of Ehsani et al. [17] and Gamlath et al. [21]. This extends the recent work of [21] to arbitrary patience in what they refer to as the *query-commit model*.

The results of this section also yield a lower bound (positive result) on the adaptivity gap of the bipartite stochastic matching problem with one-sided patience.

3.1 Defining the Probing Algorithm

We now consider the probing algorithm which is the subject of Theorems 3.1 and 3.5.

Algorithm 3 Known Stochastic Graph

```
Input $G = (U, V, E)$, a stochastic graph with edge probabilities $(p_e)_{e \in E}$, edge weights $(w_e)_{e \in E}$ and patience parameters $(\ell_v)_{v \in V}$
Set $\mathcal{M} \leftarrow \emptyset$.
Solve LP-new, and find an optimal solution $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$.
for $t = 1, \ldots, |V|$ do
    Process $v_t$ (vertex arriving at time $t$)
    Set $(u_t, v_t) \leftarrow$ VertexProbe$(G, (x_{v_t}(\boldsymbol{u}))_{\boldsymbol{u} \in U^{(\leq \ell_{v_t})}}, v_t)$.
    if $(u_t, v_t) \neq \emptyset$ and $u_t$ is unmatched then
        Set $\mathcal{M}(v_t) = u_t$.
    end if
end for
Return $\mathcal{M}$.
```

3.2 Adversarial Arrivals

We first consider the known stochastic online matching problem in the case of arbitrary patience, offline vertex weights and adversarial online vertex arrivals. Specifically, we provide a proof of Theorem 3.1.

Theorem 3.1. *If Algorithm 3 is passed a stochastic graph $G = (U, V, E)$ with offline vertex weights $(w_u)_{u \in U}$ (that is, $w_{u,v} = w_u$ for all $(u, v) \in E$) and arbitrary patience, then*

$$\mathbb{E}[val(\mathcal{M})] \geq \left(1 - \frac{1}{e}\right) \cdot OPT(G).$$

Thus, the competitive ratio of this algorithm (when the stochastic graph and order of online vertices is chosen by an adversary) is $1 - 1/e$ against the committal benchmark.

Proof. Let us now denote val$(\mathcal{M})$ as the value of the matching returned by Algorithm 3. Observe that

$$\mathbb{E}[\mathrm{val}(\mathcal{M})] = \sum_{u \in U} w_u \, \mathbb{P}[u \text{ is matched by the algorithm}].$$

As such, for each fixed $u \in U$, we may focus on lower bounding the probability that the algorithm matches it.

Recall that associated with the solution $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\ell_v)}}$ are the edge variables $(\widetilde{x}_{u,v})_{u \in U, v \in V}$, as defined following LP-new in Section 2.

Observe now that $u \in U$ is matched by the algorithm, if and only if there exists some $v \in V$ which commits to u while executing VertexProbe. For a fixed v, we denote this event by $C(u,v)$. Using Lemma 2.3, we get that

$$\mathbb{P}[C(u,v)] = p_{u,v} \, \widetilde{x}_{u,v},$$

for each $v \in V$.

Now, since the executions of VertexProbe are independent, so are the events $\{\neg C(u,v)\}_{v \in V}$, and so

$$\mathbb{P}[u \text{ is not matched}] = \prod_{v \in V} (1 - p_{u,v} \, \widetilde{x}_{u,v}).$$

As a result,

$$\begin{aligned}
\mathbb{P}[u \text{ is matched}] &= 1 - \prod_{v \in V} (1 - p_{u,v} \, \widetilde{x}_{u,v}) \\
&\geq 1 - \prod_{v \in V} \exp\left(-p_{u,v} \, \widetilde{x}_{u,v}\right) \\
&= 1 - \exp\left(- \sum_{v \in V} p_{u,v} \, \widetilde{x}_{u,v} \right),
\end{aligned}$$

as $1 - z \leq \exp(-z)$ for all $z \in \mathbb{R}$ (here we use $\exp(z) := e^z$ for notational clarity).

Now $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$ is a feasible solution to LP-new, and so

$$\sum_{v \in V} p_{u,v} \, \widetilde{x}_{u,v} \leq 1.$$

We may therefore conclude that

$$\mathbb{P}[u \text{ is matched}] \geq (1 - \exp(-1)) \sum_{v \in V} p_{u,v} \, \widetilde{x}_{u,v},$$

since $1 - \exp(-z) \geq (1 - \exp(-1))\, z$ for all $0 \leq z \leq 1$.

Thus,

$$\begin{aligned}
\mathbb{E}[\mathrm{val}(\mathcal{M})] &= \sum_{u \in U} w_u \, \mathbb{P}[u \text{ is matched}] \\
&\geq (1 - \exp(-1)) \sum_{u \in U} \sum_{v \in V} w_u \, p_{u,v} \, \widetilde{x}_{u,v} \\
&= (1 - \exp(-1)) \, \mathrm{LPOPT}_{new}(G),
\end{aligned}$$

as $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$ is an optimal solution to LP-new. By Theorem 2.2, $\mathrm{OPT}(G) \leq \mathrm{LPOPT}_{new}(G)$, and so the proof is complete.

□

Suppose that we fix an ordering π of V. We can then define $\mathrm{OPT}(G, \pi)$ to be the largest expected value an *online* probing algorithm can attain on G, provided it is presented the vertices V in the order π. With this definition, we can define the **order gap** of G as the worst case ratio between $\mathrm{OPT}(G, \pi_1)$ and $\mathrm{OPT}(G, \pi_2)$, over all orderings π_1 and π_2 of V; that is, the ratio $\mathrm{order}(G)$, where

$$\mathrm{order}(G) := \frac{\inf_{\pi_1} \mathrm{OPT}(G, \pi_1)}{\sup_{\pi_2} \mathrm{OPT}(G, \pi_2)}. \tag{3.1}$$

If $\mathscr{C}$ corresponds to the collection of all vertex weighted stochastic graphs, then we can define the **order gap** of $\mathscr{C}$ as the minimum value of $\mathrm{order}(G)$ over all $G \in \mathscr{C}$; that is, the value $\inf_{G \in \mathscr{C}} \mathrm{order}(G)$.

Since Algorithm 3 achieves a competitive ratio of $1 - 1/e$ no matter the order the online vertices are presented to it, we observe the following corollary:

Corollary 3.2. *The order gap of the collection of vertex weighted stochastic graphs is no smaller than $1 - 1/e$. Note that this is a positive result.*

We contrast this observation with an upper bound (negative result) on the order gap.

Example 3.3. *Let us consider the bipartite graph $G = (U, V, E)$ where $U = \{u_1, u_2\}$ is the set of offline vertices and $V = \{v_1, v_2\}$ is the set of online vertices each of which has unit patience. Denote $E = \{(u_1, v_1), (u_2, v_1), (u_2, v_2)\}$ as the set of edges of G, with edge probabilities $p_{u_1,v_1} = \frac{1}{2}$, $p_{u_2,v_1} = 1$ and $p_{u_2,v_2} = \frac{1}{2}$. The order gap of G is at most 0.8.*

Proof. If we process vertex v_1 before vertex v_2, we either probe (u_1, v_1) or (u_2, v_1) while processing v_1. If we probe the former, the only other edge that we can probe is (u_2, v_2) and the expected size of the constructed matching is $\frac{1}{2} + \frac{1}{2} = 1$, as both of these edges are active with probability $\frac{1}{2}$. If instead we probe edge (u_2, v_1), then it is always active and there are no more edges that we can probe. This also gives an expected matching size of 1. Thus, the maximum expected size of the matching is 1 when v_1 is processed before v_2.

If instead we process vertex v_2 before v_1, we may probe (u_2, v_2), which is active with probability $\frac{1}{2}$. If the edge is active, we next probe edge (u_1, v_1), which is active with probability $\frac{1}{2}$. In this case, expected size of matching found is $\frac{3}{2}$. If (u_2, v_2) is inactive, we instead probe (u_2, v_1), which is always active and so in this case, the expected size of matching found is 1. Thus, the maximum expected size of matching when v_2 is probed before v_1 is $\frac{1}{2}\frac{3}{2} + \frac{1}{2}1 = \frac{5}{4}$. □

While it would be interesting to know the precise value of the order gap (even just for the case of offline vertex weights), $1 - 1/e$ is the limitation of our techniques for proving positive results, as demonstrated by the following example:

Example 3.4. *Consider a graph G with a single offline node u and a collection of n online nodes V. For each edge $e = (u, v)$ with $v \in V$, set $p_{u,v} := 1/n$. As the example is in the setting of unit patience, LP-new and LP-std are equivalent, and in particular, $LPOPT_{new}(G) = LPOPT_{std}(G)$. Thus, we describe the remainder of the example with respect to the definition of LP-std for simplicity.*

Observe that the LP solution $x_{u,v} := 1$ for each $v \in V$ satisfies the constraints of LP-std. Moreover, it evaluates to an objective value of 1. Thus, $LPOPT_{std}(G) \geq 1$.

Observe now that if we consider an arbitrary probing algorithm, then its only option is to probe the edges of u in some arbitrary order (or not at all). Of course, each edge is active with probability $1/n$, so we observe that

$$\mathbb{P}[G \textit{ has a least one active edge}] = 1 - (1 - 1/n)^n = (1 + o(1))\left(1 - \frac{1}{e}\right),$$

as we allow $n \to \infty$.

As a result,

$$\inf_G \frac{OPT(G)}{LPOPT_{new}(G)} \leq \left(1 - \frac{1}{e}\right),$$

and so in particular, Algorithm 3 achieves the best possible bound against $LPOPT_{new}(G)$.

3.3 Random Order Arrivals

We now consider the known stochastic matching problem in the case of arbitrary edges weights and ROM arrivals. We first prove Theorem 3.5, which shows that Algorithm 7 gets a competitive ratio of 1/2. After arguing that the analysis is tight, we introduce a modified VertexProbe algorithm, which we refer to as VertexProbe-S. By replacing VertexProbe with the subroutine VertexProbe-S in Algorithm 3, we are able to improve the competitive guarantee to $1 - 1/e$.

Theorem 3.5. *In the ROM input model, if Algorithm 3 is passed a stochastic graph $G = (U, V, E)$ with arbitrary edge weights $(w_e)_{e \in E}$ and patience $(\ell_v)_{v \in V}$, then*

$$\mathbb{E}[val(\mathcal{M})] \geq \frac{1}{2} \cdot OPT(G),$$

Thus, against the committal benchmark, the competitive ratio of this algorithm (when the stochastic graph is chosen by an adversary and order of online vertices is determined uniformly at random) is 1/2.

We include the proof of Theorem 3.5 in Appendix F, as it has a relatively simple analysis and helped motivate the improvement to $1 - 1/e$. We now consider the following example, which confirms the performance guarantee of Algorithm 3 is tight:

Example 3.6. *Let $G = (U, V, E)$ be a bipartite graph with a single offline node u, online vertices $V = \{v_1, v_2\}$ and edges $E = \{(u, v_1), (u, v_2)\}$. We assume that the online nodes have unit patience.*

Fix $0 < \epsilon < 1$, and define the edge probabilities $p_{(u,v_1)} := \epsilon$ and $p_{u,v_2} := 1 - \epsilon$. Moreover, define the weights of the edges as $w_{u,v_1} := 1/\epsilon$ and $w_{u,v_2} = \epsilon/(1 - \epsilon)$.

For this instance, if we allow $\epsilon \to 0$, then the expected weight of matching returned by Algorithm 3 in the ROM setting is at most half that of $OPT(G)$.

Proof. Since we work in the unit patience setting for G, we express the relevant linear program as in the setting of LP-std:

$$\begin{aligned}
\text{maximize} \quad & w_{u,v_1} \cdot p_{u,v_1} \cdot x_{u,v_1} + w_{u,v_2} \cdot p_{u,v_2} \cdot x_{u,v_2} && (3.2)\\
\text{subject to} \quad & p_{u,v_1} \cdot x_{u,v_1} + p_{u,v_2} \cdot x_{u,v_2} \leq 1 && (3.3)\\
& 0 \leq x_{u,v_1} \leq 1 && (3.4)\\
& 0 \leq x_{u,v_2} \leq 1 && (3.5)
\end{aligned}$$

The optimal solution to this LP corresponds to $x_{u,v_1} = x_{u,v_2} = 1$, and the optimal value is $1 + \epsilon$.

Now, when considering the order in which v_1 arrives before v_2, if we probe the edge $e \in E$ with probability x_e, the expected value of matching returned is

$$w_{u,v_1}\, x_{u,v_1}\, p_{u,v_1} + (1 - x_{u,v_1}\, p_{u,v_1})\, w_{u,v_2}\, x_{(u,v_2)}\, p_{(u,v_2)} = 1 + (1 - \epsilon)\, \epsilon.$$

Similarly, when considering the order in which v_2 arrives before v_1, if we probe the edge $e \in E$ with probability x_e, then the expected value of matching returned is

$$w_{u,v_2}\, x_{u,v_2}\, p_{u,v_2} + (1 - x_{u,v_2}\, p_{u,v_2})\, w_{u,v_1}\, x_{u,v_1}\, p_{u,v_1} = 2\, \epsilon.$$

Thus, as the order of arrivals is determined uniformly at random, the expected value of the matching returned is

$$\frac{1}{2}(2\epsilon + 1 + \epsilon - \epsilon^2),$$

which tends to $1/2$ as ϵ tends to 0. Moreover, the optimum value of the LP tends to 1 as ϵ tends to 0, so the ratio of these values tends to $1/2$. Moreover, for this specific choice of G,

$$\text{LPOPT}_{new}(G) = \text{OPT}(G),$$

and so the proof is complete. □

We remark that there is an online algorithm that achieves the optimum expectation since the stochastic graph is known. That is, the online algorithm would simply not probe (u, v_1), if v_1 is the first arrival.

3.3.1 Improving Upon the Competitive Ratio

In [21], Gamlath et al. showed, among other things, that when $G = (U, V, E)$ has full patience, there exists a probing algorithm which achieves an approximation ratio of $1 - 1/e$. This algorithm in fact executes in the online setting, in which G is known and the probing algorithm respects a vertex order that is generated uniformly at random.

In addition to using their new LP relaxation (as discussed in Section 2), Gamlath et al. adapted the techniques of Ehsani et al. [17] from the *prophet secretary problem* to get a competitive ratio of $1 - 1/e$ in the case of full patience. We now generalize their algorithm to attain the same competitive ratio, while handling arbitrary patience constraints. We emphasize that our analysis proceeds almost identically, though this is only made possible by our definition of LP-new [13].

Given an arbitrary stochastic graph $G = (U, V, E)$, let us suppose we are presented an optimum solution to LP-new, denoted $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$, whose edge variables we denote by $(\widetilde{x}_{u,v})_{u \in U, v \in V}$. In this case, define

$$c_u := \sum_{v \in V} w_{u,v}\, p_{u,v}\, \widetilde{x}_{u,v}$$

for each $u \in U$. We can view c_u as corresponding to the contribution of u to the evaluation of $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$ as a solution to LP-new. Specifically, observe that

$$\sum_{u \in U} c_u = \sum_{u \in U, v \in V} w_{u,v}\, p_{u,v}\, \widetilde{x}_{u,v} = \text{LPOPT}_{new}(G). \tag{3.6}$$

Let us now return to the ROM setting, though we describe it in a slightly different way for the stochastic graph $G = (U, V, E)$. As in Devanur et al. [16], for each $v \in V$, draw $Y_v \in [0, 1]$ independently and uniformly at random. We assume that the vertices of V are presented to the algorithm in an increasing order, based on the values of $(Y_v)_{v \in V}$. In this way, we say that vertex $v \in V$ *arrives at time* Y_v. Observe that the vertices of V are presented to the algorithm in a uniformly at random order, so this interpretation is equivalent to the ROM setting.

We now describe a modification of Algorithm 3 that is more selective as to which of the edges returned by VertexProbe we are willing to accept. Specifically, when v arrives at time Y_v, VertexProbe is executed as before. However, if VertexProbe returns the edge $e = (u, v)$, then we only add e to the matching provided u is unmatched *and* $w_e \geq (1 - e^{Y_v - 1}) \cdot c_u$. Of course, this

[13] In Appendix A, we provide a reasonable generalization of the Gamlath et al. LP, and show that it attains the same value as LP-new.

high level description of the online probing algorithm clearly does *not* respect commitment, but fortunately we can run a *simulated* version of VertexProbe, which we refer to as VertexProbe-S.

Algorithm 4 VertexProbe-S

```
Input the stochastic graph $G = (U, V, E)$, a fixed node $s \in V$, the variables $(x_s(\boldsymbol{u}))_{\boldsymbol{u} \in U^{(\leq \ell_s)}}$
associated to $s$ in a solution to LP-new for $G$, and $0 \leq z \leq 1$.
Initialize $\mathcal{M} \leftarrow \emptyset$.
Return $\mathcal{M}$ with probability $1 - \sum_{\boldsymbol{u} \in U^{(\leq \ell_s)}} x_s(\boldsymbol{u})$        ▷ *pass* with a certain probability.
Draw $\boldsymbol{u}^*$ from $U^{(\leq \ell_s)}$ with probability $x_s(\boldsymbol{u}^*)$ (see (2.9)).
Denote $\boldsymbol{u}^* = (u_1^*, \ldots, u_k^*)$ for $k := |\boldsymbol{u}|$.
**for** $i = 1, \ldots, k$ **do**
    **if** $w_{u_i^*, s} \geq (1 - e^{z-1}) \cdot c_{u_i^*}$ **then**
        Probe $(u_i^*, s)$.
        **if** $\mathrm{st}(u_i^*, s) = 1$ **then**
            Set $\mathcal{M}(s) \leftarrow u_i^*$ and return $\mathcal{M}$.
        **end if**
    **else** draw $Z \sim \mathrm{Ber}(p_{u_i^*, v})$ independently.        ▷ a Bernoulli of parameter $p_{u_i^*, v}$.
        **if** $Z = 1$ **then**
            Set $\mathcal{M}(s) \leftarrow u_i^*$ and return $\mathcal{M}$.        ▷ drawing $Z$ simulates an edge probe.
        **end if**
    **end if**
**end for**
Return $\mathcal{M}$.
```

Remark. *Observe that* VertexProbe-S *makes a probe to the edge (u, s), only if*

$$w_{u,s} \geq (1 - e^{z-1}) \cdot c_u.$$

If this condition is not satisfied, then it still may return the edge (u, s), however (u, s) will not be probed. We make sure to return (u, s), so that VertexProbe-S *can be coupled with* VertexProbe*, as this will simplify the proof of Theorem 3.8.*

We say that VertexProbe-S **commits** to the edge (u, s), provided it returns this edge (even if it doesn't actually probe (u, s)), when executed with the parameter $0 \leq z \leq 1$. Observe that VertexProbe-S returns (u, s) with the same probability as VertexProbe, so we can make use of Lemma 2.3 to get an analogous guarantee.

Lemma 3.7. *Suppose $G = (U, V, E)$ is a stochastic graph with fixed node $s \in V$ and LP-new solution $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$, whose induced edge variables we denote by $(\widetilde{x}_{u,v})_{u \in U, v \in V}$.*

If VertexProbe-S *is passed the fixed node s, then (u, s) is returned with probability $p_{u,v} \cdot \widetilde{x}_{u,s}$ for each $u \in U$, no matter which value of $0 \leq \alpha \leq 1$ is presented to* VertexProbe-S*. Moreover, the edge (u, s) is probed only if $w_{u,s} \geq (1 - e^{z-1}) \cdot c_u$.*

We now can implement a modified version of Algorithm 3 which executes identically to the high level modification we just described, while respecting commitment.

Algorithm 5 Modified Known Stochastic Graph

```
Input $G = (U, V, E)$, a stochastic graph with edge probabilities $(p_e)_{e \in E}$, edge weights $(w_e)_{e \in E}$ and patience parameters $(\ell_v)_{v \in V}$.
Set $\mathcal{M} \leftarrow \emptyset$.
Solve LP-new, and find an optimal solution $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$.
For each $v \in V$, draw $Y_v \in [0, 1]$ independently and uniformly at random.
for $v \in V$ in increasing order of $Y_v$ do
    Set $(u, v) \leftarrow$ VERTEXPROBE-S$(G, v, (x_v(\boldsymbol{u}))_{\boldsymbol{u} \in U^{(\leq \ell_v)}}, Y_v)$.
    if $(u, v) \neq \emptyset$, and $w_{u,v} \geq (1 - e^{Y_v - 1}) \cdot c_u$ then
        if $u$ is unmatched then
            Set $\mathcal{M}(v) = u$.        ▷ $(u, v)$ is matched only if $(u, v)$ is probed and $\text{st}(u, v) = 1$.
        end if
    else
        Pass on $(u, v)$.
    end if
end for
Return $\mathcal{M}$.
```

Theorem 3.8. *In the ROM input model, if Algorithm 5 is passed a stochastic graph $G = (U, V, E)$ with arbitrary edge weights $(w_e)_{e \in E}$ and patience $(\ell_v)_{v \in V}$, then*

$$\mathbb{E}[val(\mathcal{M})] \geq \left(1 - \frac{1}{e}\right) \cdot OPT(G),$$

Thus, the competitive ratio of this algorithm is $1 - 1/e$ against the committal benchmark.

The analysis of Theorem 3.8 follows very closely the full patience proof presented in Gamlath et al. [21], and hence is mainly motivated by the single item prophet secretary problem of Ehsani et al. [17]. However, for sake of completeness, we include the argument.

Proof of Theorem 3.8. For each offline node $u \in U$, denote $\text{val}(\mathcal{M}(u))$ as the weight of the edge assigned to u (which is zero, if u remains unmatched).

Observe then that

$$\mathbb{E}[\text{val}(\mathcal{M})] = \sum_{u \in U} \mathbb{E}[\text{val}(\mathcal{M}(u))].$$

Thus, in order to complete the proof it suffices to show that

$$\mathbb{E}[\text{val}(\mathcal{M}(u))] \geq \left(1 - \frac{1}{e}\right) \cdot c_u \tag{3.7}$$

for each $u \in U$, as we know that $\sum_{u \in U} c_u = \text{LPOPT}_{new}(G) \geq \text{OPT}(G)$ (by Theorem 2.2).

As such, let us suppose $u \in U$ is fixed for the remainder of the proof. The remaining computations follow Ehsani et al. [17] for the single item prophet secretary problem, though we must make use Lemma 3.7 and constraint (2.8) of LP-new.

Let us now define the random variables N_u and $M_{u,v}$ where:

1. $N_u := \sum_{v \in V} (1 - e^{Y_v - 1}) \cdot c_u \cdot \mathbf{1}_{[\mathcal{M}(u) = v]}$,

2. $M_{u,v} := \big(w_{u,v} - (1 - e^{Y_v - 1}) \cdot c_u\big) \cdot \mathbf{1}_{[\mathcal{M}(u) = v]}$.

That is, if u is matched to v, then N_u is assigned the value $(1 - e^{Y_v - 1}) \cdot c_u$ and $M_{u,v}$ is assigned $\big(w_{u,v} - (1 - e^{Y_v - 1}) \cdot c_u\big)$. Proving (3.7) thus reduces to showing that

$$\mathbb{E}[N_u] + \sum_{v \in V} \mathbb{E}[M_{u,v}] \geq \left(1 - \frac{1}{e}\right) \cdot c_u.$$

In other words, the cumulative amount assigned to u and the vertices of V is at least $(1 - 1/e) \cdot c_u$ in expectation[14].

We first focus on lower bounding $\mathbb{E}[N_u]$. Let us define T_u as the arrival time of the online vertex which matches to u (which is 1, if no such vertex exists). For convenience, define the functions $r = r(t), F = F(t)$ and $\alpha = \alpha(t)$, where for $t \in [0, 1]$.

$$r(t) := \mathbb{P}[T_u \geq t], \; F(t) := 1 - r(t), \text{ and } \alpha(t) := 1 - e^{t-1}.$$

Define $C(u, v)$ as the event in which v commits to u when VertexProbe-S is executed using $v \in V$. Observe that the events $\{C(u, v)\}_{v \in V}$ are independent, as the executions of VertexProbe-S in Algorithm 5 are themselves independent.

For each $v \in V$, let us now define the function $\psi_{u,v} = \psi_{u,v}(t)$ where $\psi_{u,v}(t) := \mathbb{P}[Y_v < t \text{ and } w_{u,v} \geq \alpha(t) \cdot c_u]$ for $t \in [0, 1]$. If we now apply Lemma 3.7, and additionally make use of the independence of the variables $(Y_v)_{v \in V}$, then we may conclude that

$$\begin{aligned} r(t) &= \mathbb{P}[T_u \geq t] \\ &= 1 - \prod_{v \in V} (1 - \psi_{u,v}(t) \cdot \mathbb{P}[C(u, v)]) \\ &= 1 - \prod_{v \in V} (1 - \psi_{u,v}(t) \cdot \widetilde{x}_{u,v}) \end{aligned}$$

for each $t \in [0, 1]$.

Thus, since the functions $(\psi_{u,v})_{v \in V}$ are clearly all continuously differentiable on $(0, 1)$, r is continuously differentiable as well. On the other hand, observe that F is the c.d.f of T_u, so we have that

$$\begin{aligned} \mathbb{E}[N_u] &= \int_0^1 \alpha(t) \cdot c_u \cdot dF(t) \\ &= \int_0^1 \alpha(t) \cdot c_u \cdot F'(t) \, dt \\ &= -\int_0^1 \alpha(t) \cdot c_u \cdot r'(t) \, dt \end{aligned}$$

Thus,

$$\mathbb{E}[N_u] = -\int_0^1 \alpha(t) \cdot c_u \cdot r'(t) \, dt,$$

and so we may apply integration by parts to get that

$$\begin{aligned} \int_0^1 \alpha(t) \cdot c_u \cdot r'(t) \, dt &= c_u \left([r(t) \cdot \alpha(t)]_{t=0}^1 - \int_0^1 r(t) \cdot \alpha'(t) \, dt \right) \\ &= c_u \left((1 - 1/e) + \int_0^1 r(t) \cdot \alpha'(t) \, dt \right). \end{aligned}$$

[14] Ehsani et al. [17] and Gamlath et al. [21] provide a utility/revenue interpretation of these variables.

To conclude,

$$\mathbb{E}[N_u] = c_u \left((1 - 1/e) + \int_0^1 r(t) \cdot \alpha'(t) \, dt \right). \tag{3.8}$$

Let us now focus on lower bounding $\sum_{v \in V} \mathbb{E}[M_{u,v}]$. First observe that if $0 \leq t \leq 1$ satisfies $w_{u,v} \geq \alpha(t) \cdot c_u$, then

$$\mathbb{E}[M_{u,v} \,|\, Y_v = t] = p_{u,v} \, \widetilde{x}_{u,v} \cdot (w_{u,v} - \alpha(t) \cdot c_u) \cdot \mathbb{P}[T_u \geq t \,|\, Y_v = t],$$

as v is matched to u with probability $p_{u,v} \, \widetilde{x}_{u,v}$, given u is unmatched at time t. Moreover, if $0 \leq t \leq 1$ satisfies $w_{u,v} < \alpha(t) \cdot c_u$, then $\mathbb{E}[M_{u,v} \,|\, Y_v = t] = 0$. Thus, for all $0 \leq t \leq 1$,

$$\mathbb{E}[M_{u,v} \,|\, Y_v = t] \geq p_{u,v} \, \widetilde{x}_{u,v}(w_{u,v} - \alpha(t) \cdot c_u) \cdot \mathbb{P}[T_u \geq t \,|\, Y_v = t]. \tag{3.9}$$

On the other hand, it is clear that $\mathbb{P}[T_u \geq t \,|\, Y_v = t] \geq \mathbb{P}[T_u \geq t]$. Thus, after applying (3.9) and observing $r(t) = \mathbb{P}[T_u \geq t]$, we get that

$$\begin{aligned} \sum_{v \in v} \mathbb{E}[M_{u,v} \,|\, Y_v = t] &\geq \sum_{v \in V} p_{u,v} \, \widetilde{x}_{u,v} \cdot (w_{u,v} - \alpha(t) \cdot c_u) \cdot r(t) \\ &= \left(c_u - \alpha(t) \cdot c_u \cdot \sum_{v \in V} p_{u,v} \, \widetilde{x}_{u,v} \right) \cdot r(t). \end{aligned}$$

Now, we know that $\sum_{v \in V} p_{u,v} \, \widetilde{x}_{u,v} \leq 1$ by constraint (2.8) of LP-new. Thus,

$$\sum_{v \in v} \mathbb{E}[M_{u,v} \,|\, Y_v = t] \geq c_u \cdot (1 - \alpha(t)) \cdot r(t),$$

and so since the random variables $(Y_v)_{v \in V}$ are uniformly distributed, we get that

$$\sum_{v \in v} \mathbb{E}[M_{u,v}] \geq c_u \cdot \int_0^1 (1 - \alpha(t)) \cdot r(t) \, dt.$$

By combining this equation with (3.8), it follows that

$$\begin{aligned} \mathbb{E}[N_u] + \sum_{v \in v} \mathbb{E}[M_{u,v}] &\geq c_u \cdot \left((1 - 1/e) + \int_0^1 r(t) \cdot \alpha'(t) \, dt \right) + c_u \cdot \int_0^1 (1 - \alpha(t)) \cdot r(t) \, dt \\ &= c_u \cdot (1 - 1/e) + c_u \int_0^1 r(t) \cdot (1 - \alpha(t) + \alpha'(t)) \, dt \\ &= (1 - 1/e) \cdot c_u, \end{aligned}$$

as $1 - \alpha(t) + \alpha'(t) = 0$ for all $t \in [0, 1]$.

As this argument holds for each $u \in U$, the proof is complete.

□

4 Online Stochastic Matching in the Known I.I.D. Model

In this section, we consider a generalization of the classical known i.i.d. matching problem (introduced in Feldman et al. [19]) to the stochastic matching setting (as first studied in Bansal et al. [6]). Once we review the relevant framework and terminology, we introduce an online probing algorithm which achieves a competitive ratio of $1 - 1/e$ for arbitrary patience and edge weights, thereby proving Theorem 4.3. Our algorithm generalizes the algorithm of [10] to arbitrary patience, in which Brubach et al. proved a competitive ratio of $1 - 1/e$ for the unit patience setting. This allows us to improve upon the previously best known competitive ratio of 0.46 for the case of arbitrary patience, as presented in [11].

We again note that our positive results from Section 3 do *not* seem to immediately imply the same positive results in the known i.i.d. stochastic matching problem, as we explain in more detail after reviewing the model.

4.1 The Known I.I.D. Stochastic Setting

Let us suppose that $G = (U, V, E)$ is a stochastic graph with edges weights $(w_e)_{e \in E}$, edge probabilities $(p_e)_{e \in E}$ and offline patience values $(\ell_v)_{v \in V}$ associated with it. In the known i.i.d. setting, we refer to G as a **stochastic type graph** (or **type graph** when clear), and the vertices of V as the **type nodes** of G.

Now, fix a parameter $n \geq 1$ (which need not be equal to $|V|$), indicating the number of **rounds** or **arrivals** to occur. Moreover, consider $\boldsymbol{r} = (r_v)_{v \in V}$, where $r_v > 0$ for each $v \in V$, and $\sum_{v \in V} r_v = n$. We refer to r_v as the (fractional) **arrival rate** of type node $v \in V$. An input to the stochastic known i.i.d. matching problem then consists of the tuple $(G, \boldsymbol{r}, n)$, which we refer to as a **known i.i.d. input** with fractional arrival rates.

An **online probing algorithm** $\mathcal{A}$ is given access to $(G, \boldsymbol{r}, n)$ as part of its input. For each $t = 1, \ldots, n$, *vertex arrival* $v_t \in V$ is drawn independently in round t using the distribution $\boldsymbol{r}/n$, at which point v_t is said to be of type $v \in V$, provided $v_t = v$. We emphasize that the edge states of v_t are statistically independent from the edge states of all the previously drawn nodes (even if v_t is not the first vertex of type v to arrive).

Using all past available information regarding the outcomes of the probes involving $v_1, \ldots, v_{t-1}$, together with the edge probabilities $(p_{u,v_t})_{u \in U}$, weights $(w_{u,v_t})_{u \in U}$ and patience value ℓ_{v_t}, $\mathcal{A}$ may probe up to ℓ_{v_t} edges adjacent to v_t. The algorithm is again restricted by commitment, in that v_t may only be matched to the first $u \in U$ for which the probe to (u, v_t) confirms that the edge is active.

Observe that while the type graph $(G, \boldsymbol{r}, n)$ is passed as input to $\mathcal{A}$, the stochastic graph $\mathcal{A}$ actually executes on is in fact randomly generated, and unknown to $\mathcal{A}$. Let us denote this (random) stochastic graph by $\hat{G} = (U, \hat{V}, \hat{E})$. Here, $\hat{V}$ consists of the random arrival nodes of V presented to the algorithm, and $\hat{E}$ includes all the relevant edges between U and $\hat{V}$ (since the same node from V can arrive multiple times, $\hat{V}$ and $\hat{E}$ are multisets). We assume that $\hat{G}$ also encodes all the edge weights, probabilities and patience values induced from the arrival nodes of $\hat{V}$.

We refer to $\hat{G}$ as the **instantiated stochastic graph** or simply the **instantiated graph** when clear. Observe that since $(G, \boldsymbol{r}, n)$ encodes the distribution of $\hat{G}$, we say that $\hat{G}$ is distributed according to the known i.i.d. input $(G, \boldsymbol{r}, n)$, which we denote by $\hat{G} \sim (G, \boldsymbol{r}, n)$.

Denote $\text{val}(\mathcal{A}(\hat{G}))$ as the value of the matching $\mathcal{A}$ constructs when passed the instantiated graph $\hat{G}$. Our performance measure for $\mathcal{A}$ then involves averaging over all the possible instantiations of $\hat{G}$. Specifically, we wish to maximize

$$\mathbb{E}[\text{val}(\mathcal{A}(\hat{G}))],$$

where the expectation is over the randomness in drawing $\hat{G}$ from $(G, \boldsymbol{r}, n)$, together with the inherent randomness in the states of the edges of $\hat{G}$, as well as any randomized decisions $\mathcal{A}$ may make.

For each randomly drawn $\hat{G} \sim (G, \boldsymbol{r}, n)$, we can consider the committal benchmark, and the evaluation it takes on $\hat{G}$, namely $\text{OPT}(\hat{G})$. This yields a committal probing strategy, which we refer to as the **committal benchmark** for the stochastic type graph $(G, \boldsymbol{r}, n)$. We denote the expected performance of the committal benchmark by $\mathrm{OPT}(G, \boldsymbol{r}, n)$. Observe that

$$\mathrm{OPT}(G, \boldsymbol{r}, n) = \mathbb{E}[\mathrm{OPT}(\hat{G})],$$

where the expectation is over the randomness in generating $\hat{G}$. We can define the **non-committal benchmark** for the stochastic type graph $(G, \boldsymbol{r}, n)$ analogously, which we denote by $\mathrm{OPT}_{non}(G, \boldsymbol{r}, n)$.

The standard in the literature (see [3, 6, 11]) is to prove competitive ratios against the committal benchmark. More precisely, the goal is to find an online probing algorithm $\mathcal{A}$ for which the (strict) competitive ratio

$$\inf_{(G,\boldsymbol{r},n)} \frac{\mathbb{E}[\mathrm{val}(\mathcal{A}(\hat{G}))]}{\mathrm{OPT}(G, \boldsymbol{r}, n)}$$

is as close to 1 as possible.

Before continuing, we emphasize that there does *not* seem to be an obvious reduction from the known i.i.d. stochastic matching problem to the known stochastic matching problem with ROM arrivals. Specifically, suppose we are presented an online probing algorithm $\mathcal{A}$ which achieves competitive ratio $0 < c \leq 1$ in the known stochastic matching problem with ROM arrivals. In this case, let us now fix a stochastic type graph $(G, \boldsymbol{r}, n)$, and imagine trying to use $\mathcal{A}$ to design a probing algorithm for the i.i.d. matching problem. If we consider the instantiated graph $\hat{G}$ drawn from $(G, \boldsymbol{r}, n)$, then the online vertices of $\hat{G}$ will indeed be presented to $\mathcal{A}$ in a random order. That being said, in order for $\mathcal{A}$ to attain to attain a competitive guarantee of $c \cdot \mathrm{OPT}(\hat{G})$, it needs to be presented the entire description of $\hat{G}$ as well. However, an online probing algorithm in the known i.i.d. setting is only given access to the type graph, $(G, \boldsymbol{r}, n)$, *not* the instantiated graph $\hat{G}$. Moreover, $\hat{G} = (U, \hat{V}, \hat{E})$ may be a very different stochastic graph than G; for instance, type node $v \in V$ may appear multiple times in $\hat{V}$, or perhaps not at all. As such, it is unclear how to modify $\mathcal{A}$ to obtain the same competitive ratio of c against $\mathrm{OPT}(G, \boldsymbol{r}, n)$.

4.2 Defining an LP Relaxation

Given an input $(G, \boldsymbol{r}, n)$ to the known i.i.d. matching problem, it is challenging to directly compare the performance of an online probing algorithm to that of the committal benchmark; that is, the value $\mathrm{OPT}(G, \boldsymbol{r}, n)$. Instead, we once again focus on LP based approaches for upper bounding this quantity.

Let us now review the LP introduced in [6, 11], as defined for $(G, \boldsymbol{r}, n)$, specialized to the case of one-sided patience.

$$\begin{aligned}
\text{maximize} \quad & \sum_{u \in V, v \in V} w_{u,v}\, p_{u,v}\, y_{u,v} && && \text{(LP-std-iid)} \\
\text{subject to} \quad & \sum_{v \in V} p_{u,v}\, y_{u,v} \leq 1 && \forall u \in U && (4.1) \\
& \sum_{u \in U} p_{u,v}\, y_{u,v} \leq r_v && \forall v \in V && (4.2) \\
& \sum_{u \in U} y_{u,v} \leq r_v \cdot \ell_v && \forall v \in V && (4.3) \\
& 0 \leq y_{u,v} \leq r_v && \forall u \in U, v \in V && (4.4)
\end{aligned}$$

If $\text{LPOPT}_{std-iid}(G, \boldsymbol{r}, n)$ denotes the value of the optimal solution to LP-std-iid, then it was shown by Bansal et al. [6] to be a relaxation of the committal benchmark; that is,

$$\text{OPT}(G, \boldsymbol{r}, n) \leq \text{LPOPT}_{std-iid}(G, \boldsymbol{r}, n).$$

Unfortunately, LP-std-iid suffers the same issues as LP-std, as Example 2.1 continues to apply, as can be seen by setting $r_v = 1$ for $v \in V$ and $n = |V|$. As such, we introduce a new LP for $(G, \boldsymbol{r}, n)$, using the same ideas as in the derivation of LP-new. The essential difference in this LP being that we incorporate the arrival rates of $(G, \boldsymbol{r}, n)$, as can be seen below in constraint (4.6).

$$\begin{aligned}
&\text{maximize} && \sum_{v \in V} \sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} \left(\sum_{i=1}^{|\boldsymbol{u}|} w_{u_i,v} \, g_v^i(\boldsymbol{u}) \right) y_v(\boldsymbol{u}) && && \text{(LP-new-iid)} \\
&\text{subject to} && \sum_{v \in V} \sum_{i=1}^{\ell_v} \sum_{\substack{\boldsymbol{u}^* \in U^{(\leq \ell_v)} \\ u_i^* = u}} g_v^i(\boldsymbol{u}^*) \, y_v(\boldsymbol{u}^*) \leq 1 && \forall u \in U && (4.5) \\
& && \sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} y_v(\boldsymbol{u}) \leq r_v && \forall v \in V, && (4.6) \\
& && y_v(\boldsymbol{u}) \geq 0 && \forall v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)} && (4.7)
\end{aligned}$$

Given a feasible solution to LP-new-iid, say $(y_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$ we define the **induced edge variables**, hereby denoted $(\widetilde{y}_{u,v})_{u \in U, v \in V}$, as in the case of LP-new; that is,

$$\widetilde{y}_{u,v} := \sum_{i=1}^{\ell_v} \sum_{\substack{\boldsymbol{u}^* \in U^{(\leq \ell_v)}: \\ u_i^* = u}} \frac{g_v^i(\boldsymbol{u}^*) \, y_v(\boldsymbol{u})}{p_{u,v}}.$$

Let us now denote $\text{LPOPT}_{new-iid}(G, \boldsymbol{r}, n)$ as the value of an optimal solution to LP-new-iid. We claim that LP-new-iid is a relaxation of the committal benchmark. This follows from a conditioning argument involving an application of Theorem 2.2 to $\hat{G} \sim (G, \boldsymbol{r}, n)$, so we defer the details to Appendix F. We remark that the techniques used in this proof constitute a general method for extending LP relaxations to the stochastic known i.i.d. setting, and so they may be of independent interest.

Lemma 4.1. *For any input $(G, \boldsymbol{r}, n)$ of the known i.i.d stochastic matching problem,*

$$OPT(G, \boldsymbol{r}, n) \leq LPOPT_{new-iid}(G, \boldsymbol{r}, n).$$

The procedure for solving LP-new-iid efficiently follows the same approach as that of LP-new, so we defer the details to Appendix B.

4.3 Defining a Known I.I.D. Probing Algorithm

We now consider an online probing algorithm for the known i.i.d. stochastic matching problem, which generalizes the unit patience probing algorithm of Brubach et al. [10].

Given $(G, \boldsymbol{r}, n)$, suppose that we consider a feasible solution to LP-new-iid, which we denote by $(y_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$. If we fix $v \in V$, then the values $(y_v(\boldsymbol{u})/r_v)_{\boldsymbol{u} \in U^{(\leq \ell_v)}}$ satisfy,

$$\sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} \frac{y_v(\boldsymbol{u})}{r_v} \leq 1, \tag{4.8}$$

as a result of constraint (4.6). As such, given the input $(U, (y_v(\boldsymbol{u})/r_v)_{\boldsymbol{u} \in U^{(\leq \ell_v)}}, v)$ for a fixed $v \in V$, we can execute VertexProbe. In particular, we can apply Lemma 2.3 in the known i.i.d. setting to get the following lemma:

Lemma 4.2. *Fix $u \in U$ and $v \in V$. For each $t = 1, \dots, n$, denote $C(u, v_t)$ as the event in which Algorithm 6 commits v_t to u in one of its probes. In this case,*

$$\mathbb{P}[C(u, v_t) \mid v_t = v] = \frac{\widetilde{y}_{u,v} \, p_{u,v}}{r_v},$$

where $(\widetilde{y}_{u,v})_{u \in U, v \in V}$ are the induced edge variables of the solution $(y_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$.

Proof. When Algorithm 6 processes v_t, it executes VertexProbe$(U, (y_{v_t}(\boldsymbol{u}^*)/r_{v_t})_{\boldsymbol{u} \in U^{(\leq \ell_{v_t})}}, v_t)$. If we condition on the event in which $v_t = v$, then this corresponds to executing VertexProbe using the input $(U, (y_v(\boldsymbol{u}^*)/r_v)_{\boldsymbol{u} \in U^{(\leq \ell_v)}}, v)$. As a result, an application of Lemma 2.3 ensures that

$$\mathbb{P}[C(u, v_t) \mid v_t = v] = \frac{\widetilde{y}_{u,v} \, p_{u,v}}{r_v},$$

thus completing the proof.

□

We now adapt Algorithm 3 to the known i.i.d. setting, leading to the following algorithm:

Algorithm 6 Known I.I.D.

```
Input $G = (U, V, E)$, an arbitrary stochastic type graph.
Input $n \geq 1$, the number of arriving vertices, and the arrivals rates of $V$, $\boldsymbol{r} = (r_v)_{v \in V}$.
Set $\mathcal{M} \leftarrow \emptyset$.
Solve LP-new-iid, and find an optimal solution $(y_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$.
for $t = 1, \dots, n$ do
    Let $v_t$ be the vertex that arrives at time t.
    Identify the type of $v_t$ in $V$, and the corresponding values $(y_{v_t}(\boldsymbol{u}^*)/r_{v_t})_{\boldsymbol{u}^* \in U^{(\leq \ell_{v_t})}}$
    Set $(u_t, v_t) \leftarrow$ VertexProbe$(U, (y_{v_t}(\boldsymbol{u}^*)/r_{v_t})_{\boldsymbol{u}^* \in U^{(\leq \ell_{v_t})}}, v_t)$.
    if $(u_t, v_t) \neq \emptyset$ and $u_t$ is unmatched then
        Set $\mathcal{M}(v_t) = u_t$.
    end if
end for
Return $\mathcal{M}$.
```

Theorem 4.3. *Algorithm 6 achieves a competitive ratio of $1 - 1/e$ against the committal benchmark, for arbitrary edge weights and patience values.*

Proof. Let us fix $u \in U$ and $v \in V$, where $G = (U, V, E)$. While Algorithm 6 executes on the instantiated graph $\hat{G} = (U, \hat{V}, \hat{E})$, let us say that the algorithm matches the edge $e = (u, v) \in E$, provided there exists some $1 \leq t \leq n$ for which $v_t = v$ and $\mathcal{M}(v_t) = u$ (here $v_1, \dots, v_n$ are the ordered arrivals of the vertices of $\hat{V}$). Observe then that

$$\mathbb{E}[\text{val}(\mathcal{M})] = \sum_{e \in E} w_e \, \mathbb{P}[e \text{ is matched}].$$

As such, we focus on lower bounding $\mathbb{P}[e \text{ is matched}]$ for each $e \in E$.

Observe now that

$$\mathbb{P}[e \text{ is matched}] = \sum_{t=1}^{n} \mathbb{P}[\mathcal{M}(v_t) = u \,|\, v_t = v] \cdot \mathbb{P}[v_t = v].$$

Moreover, if $R_t \subseteq U$ denotes the unmatched vertices of U after vertices $v_1, \ldots, v_{t-1}$ arrive, then

$$\begin{aligned}\mathbb{P}[\mathcal{M}(v_t) = u \,|\, v_t = v] &= \mathbb{P}[C(u, v_t) \cap \{u \in R_t\} \,|\, v_t = v] \\ &= \mathbb{P}[C(u, v_t) \,|\, v_t = v] \cdot \mathbb{P}[u \in R_t \,|\, v_t = v],\end{aligned}$$

as the events $C(u, v_t)$ and $u \in R_t$ are conditionally independent given $v_t = v$, since the algorithm decides upon the probes of v_t independently from those of $v_1, \ldots, v_{t-1}$.

Moreover, the event $u \in R_t$ can be determined from the probes of the vertices $v_1, \ldots, v_{t-1}$, and is therefore independent from the event $v_t = v$. Thus,

$$\mathbb{P}[\mathcal{M}(v_t) = u \,|\, v_t = v] = \mathbb{P}[C(u, v_t) \,|\, v_t = v] \cdot \mathbb{P}[u \in R_t],$$

and so

$$\mathbb{P}[\mathcal{M}(v_t) = u \,|\, v_t = v] = \widetilde{y}_{u,v}\, p_{u,v}\, \mathbb{P}[u \in R_t],$$

after applying Lemma 4.2.

It suffices to lower bound $\mathbb{P}[u \in R_t]$. Observe that for each $k = 1, \ldots, n-1$,

$$\mathbb{P}[u \in R_{k+1}] = \mathbb{P}[\cap_{j=1}^{k} \neg C(u, v_j)] = \mathbb{P}[\neg C(u, v_k)] \cdot \mathbb{P}[u \in R_k]$$

as the probes of v_k are drawn independently from those of $v_1, \ldots, v_{k-1}$.

Yet,

$$\begin{aligned}\mathbb{P}[C(u, v_k)] &= \sum_{v \in V} \mathbb{P}[C(u, v_k) \,|\, v_k = v] \cdot \mathbb{P}[v_k = v] \\ &= \sum_{v \in V} \frac{\widetilde{y}_{u,v}\, p_{u,v}}{r_v} \frac{r_v}{n} \\ &= \sum_{v \in V} \frac{\widetilde{y}_{u,v}\, p_{u,v}}{n} \\ &\leq \frac{1}{n},\end{aligned}$$

by Lemma 4.2 and the constraints of LP-new-iid. Thus,

$$\mathbb{P}[u \in R_t] \geq \left(1 - \frac{1}{n}\right)^{t-1} \tag{4.9}$$

after applying the above recursion.

As a result,

$$\mathbb{P}[\mathcal{M}(v_t) = u \,|\, v_t = v] \geq p_{u,v}\, \widetilde{y}_{u,v} \left(1 - \frac{1}{n}\right)^{t-1},$$

and so

$$\mathbb{P}[(u, v) \text{ is matched}] = \sum_{t=1}^{n} \mathbb{P}[\mathcal{M}(v_t) = u \,|\, v_t = v] \cdot \mathbb{P}[v_t = v]$$

$$
\begin{aligned}
&\geq \sum_{t=1}^{n} \frac{\widetilde{y}_{u,v}\, p_{u,v}}{r_v} \left(1 - \frac{1}{n}\right)^{t-1} \frac{r_v}{n} \\
&= \sum_{t=1}^{n} \left(1 - \frac{1}{n}\right)^{t-1} \frac{\widetilde{y}_{u,v}\, p_{u,v}}{n}.
\end{aligned}
$$

Now, $\sum_{t=1}^{n} \frac{1}{n}\left(1 - \frac{1}{n}\right)^{t-1} \geq 1 - \frac{1}{e}$, so

$$\mathbb{P}[(u,v) \text{ is matched}] \geq \left(1 - \frac{1}{e}\right) \widetilde{y}_{u,v}\, p_{u,v}$$

for each $u \in U, v \in V$. As such

$$\mathbb{E}[\mathrm{val}(\mathcal{M})] \geq \sum_{u \in U, v \in V} w_{u,v}\, \widetilde{y}_{u,v}\, p_{u,v} \left(1 - \frac{1}{e}\right).$$

Since $(y_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$ is an optimum solution to LP-new-iid, the algorithm is $1-1/e$ competitive by Lemma 4.1, thus completing the proof. □

5 Online Stochastic Matching with ROM Arrivals: The Case of an Unknown Stochastic Graph

In this section, we consider the unknown stochastic matching problem in the setting of arbitrary edge weights. Specifically, we employ the LP based techniques of the previous section to design a randomized probing algorithm which generalizes the approach of Kesselheim et al. [29]. As in [29], we make the added assumption that the number of vertex arrivals is known to the online probing algorithm ahead of time. We are then able to prove a best possible asymptotic competitive ratio of $1/e$, though unlike the work of Kesselheim et al. [29], our online algorithm requires randomization.

Let us suppose that $G = (U, V, E)$ is a stochastic graph with arbitrary edge weights, probabilities and patience values. We assume that $n := |V|$, and that the online nodes of V are denoted $v_1, \ldots, v_n$, where the order is generated uniformly at random.

Since G is unknown to us in the current setting, we cannot directly solve LP-new to define a probing algorithm. As such, we must adjust which LP we attempt to solve.

Let us suppose that S is a non-empty subset of the nodes of V. We can then denote $G[S]$ as the **induced stochastic graph** of G on S. This is constructed by taking the induced graph of G on the partite sets U and S, and restricting the edge weights and probabilities to $(p_{u,s})_{u \in U, s \in S}$ and $(w_{u,s})_{u \in U, s \in S}$ respectively, as well as the patience values to $(\ell_s)_{s \in S}$.

From now on, denote V_t as the set of first t arrivals of V; that is, $V_t := \{v_1, \ldots, v_t\}$. Moreover, set $G_t := G[V_t]$, and $\mathrm{LPOPT}_{new}(G_t)$ as the value of an optimum solution to LP-new (this is a random variable, as V_t is a random subset of V). The following inequality then holds:

Lemma 5.1. *For each $t \geq 1$,*

$$\mathbb{E}[LPOPT_{new}(G_t)] \geq \frac{t}{n}\, LPOPT_{new}(G).$$

In light of this observation, we design an online probing algorithm which makes use of V_t, the currently known nodes, to derive an optimum LP solution with respect to G_t. As such, each time an online node arrives, we must compute an optimum solution for the LP associated to G_t, distinct from the solution computed for that of G_{t-1}.

Algorithm 7 Unknown Stochastic Graph ROM

```
Input $U$, $n := |V|$, and $0 \leq \alpha \leq 1$.
Set $\mathcal{M} \leftarrow \emptyset$.
Set $G_0 = (U, \emptyset, \emptyset)$
for $t = 1, \ldots, |V|$ do
    Input $v_t$, with $(w_{u,v_t})_{u \in U}$, $(p_{u,v_t})_{u \in U}$ and $\ell_{v_t}$.
    Compute $G_t$, by updating $G_{t-1}$ to contain $v_t$ and its edges into $U$, as well its edge weights, probabilities and patience.
    if  $t < |V|\alpha$ then
        Pass on $v_t$.
    else
        Solve LP-new for $G_t$ and find an optimum solution.
        Encode this (new) optimum solution as $(x_v(\boldsymbol{u}))_{v \in V_t, \boldsymbol{u} \in U^{(\leq \ell_v)}}$.
        Process $v_t$, and set $(u_t, v_t) \leftarrow$ VertexProbe$(G_t, (x_{v_t}(\boldsymbol{u}))_{\boldsymbol{u} \in U^{(\leq \ell_{v_t})}}, v_t)$.
        if $(u_t, v_t) \neq \emptyset$ and $u_t$ is unmatched then
            Set $\mathcal{M}(v_t) = u_t$.
        end if
    end if
end for
Return $\mathcal{M}$.
```

Theorem 5.2. *When α is set to $1/e$, Algorithm 7 achieves an asymptotic competitive ratio*[15] *of $1/e$ against the committal benchmark.*

Proof. Observe that by definition, Algorithm 7 does not probe any of the neighbours of v_t for $1 \leq t \leq \alpha n - 1$. As such, these online vertices do not contribute to the matching returned by the algorithm, and so we hereby fix t and assume that $t \geq \alpha n$. We emphasize that the value of $x_v(\boldsymbol{u})$ corresponds to this fixed value of t, for each $v \in V_t$ and $\boldsymbol{u} \in U^{(\leq \ell_v)}$.

Let us now define $e_t := (u_t, v_t)$, where u_t is the vertex $u \in U$ which v_t commits to (recall that $(u_t, v_t) = \emptyset$ if v_t remains uncommitted after its probes). We now define the random variable

$$\text{val}(e_t) := w_{e_t} \mathbf{1}_{[e_t \neq \emptyset]},$$

which indicates the weight of the edge v_t commits to (which is zero, provided v_t remains uncommitted).

For each $u \in U$, denote $C(u, v_t)$ as the event in which v_t commits to u. Let us now condition on the random subset V_t, as well as the random vertex v_t. In this case,

$$\mathbb{E}[\text{val}(e_t) \,|\, V_t, v_t] = \sum_{u \in U} w_{u,v_t} \, \mathbb{P}[C(u, v_t) \,|\, V_t, v_t].$$

Observe however that once we condition on V_t and v_t, Algorithm 7 corresponds to executing VertexProbe on the instance $(G_t, (x_{v_t}(\boldsymbol{u}))_{\boldsymbol{u} \in U^{(\leq \ell_{v_t})}}, v_t)$. Thus, Lemma 2.3 implies that

$$\mathbb{P}[C(u, v_t) \,|\, V_t, v_t] = p_{u,v_t} \, \widetilde{x}_{u,v_t},$$

where $\widetilde{x}_{u,v_t}$ is the induced edge variable associated with the solution $(x_v(\boldsymbol{u}))_{v \in V_t, \boldsymbol{u} \in U^{(\ell_v)}}$. As such,

$$\mathbb{E}[\text{val}(e_t) \,|\, V_t, v_t] = \sum_{u \in U} w_{u,v_t} \, p_{u,v_t} \, \widetilde{x}_{u,v_t}.$$

[15] The asymptotic competitive ratio for an online probing algorithm $\mathcal{A}$ in the ROM setting is defined as $\liminf_{\text{OPT}(G) \to \infty} \frac{\mathbb{E}[\text{val}(\mathcal{A}(G))]}{\text{OPT}(G)}$.

On the other hand, if we condition on *solely* V_t, then v_t remains distributed uniformly at random amongst the vertices of V_t. Moreover, once we condition on V_t, the graph G_t is determined, and thus so are the values $(x_v(\boldsymbol{u}))_{v \in V_t, \boldsymbol{u} \in U^{(\ell_v)}}$ of LP-new. These observations together imply that

$$\mathbb{E}[w_{u,v_t}\, p_{u,v_t}\, \widetilde{x}_{u,v_t} \mid V_t] = \frac{\sum_{v \in V_t} w_{u,v}\, p_{u,v}\, \widetilde{x}_{u,v}}{t} \tag{5.1}$$

for each $u \in U$ and $\alpha n \leq t \leq n$.

If we now take expectation over v_t, then using the law of iterated expectations,

$$\begin{aligned}\mathbb{E}[\text{val}(e_t) \mid V_t] &= \mathbb{E}[\,\mathbb{E}[\text{val}(e_t) \mid V_t, v_t]\, \mid V_t] \\ &= \mathbb{E}\left[\sum_{u \in U} w_{u,v_t}\, p_{u,v_t}\, \widetilde{x}_{u,v_t} \mid V_t\right] \\ &= \sum_{u \in U} \mathbb{E}[w_{u,v_t}\, p_{u,v_t}\, \widetilde{x}_{u,v_t} \mid V_t] \\ &= \sum_{u \in U} \sum_{v \in V_t} \frac{w_{u,v} p_{u,v}\, \widetilde{x}_{u,v}}{t},\end{aligned}$$

where the final equation follows from (5.1).

Observe however that

$$\text{LPOPT}_{new}(G_t) = \sum_{v \in V_t} \sum_{u \in U} w_{u,v_t}\, p_{u,v_t}\, \widetilde{x}_{u,v_t},$$

as $(x_v(\boldsymbol{u}))_{v \in V_t, \boldsymbol{u} \in U^{(\leq \ell_{v_t})}}$ is an optimum solution to LP-new for G_t. As a result,

$$\mathbb{E}[\text{val}(e_t) \mid V_t] = \frac{\text{LPOPT}_{new}(G_t)}{t},$$

and so

$$\mathbb{E}[\text{val}(e_t)] = \frac{\mathbb{E}[\text{LPOPT}_{new}(G_t)]}{t},$$

after taking taking expectation over V_t.

On the other hand, Lemma 5.1 implies that

$$\frac{\mathbb{E}[\text{LPOPT}_{new}(G_t)]}{t} \geq \frac{\text{LPOPT}_{new}(G)}{n}.$$

Thus,

$$\mathbb{E}[\text{val}(e_t)] \geq \frac{\text{LPOPT}_{new}(G)}{n}, \tag{5.2}$$

provided $\alpha n \leq t \leq n$.

Let us now consider the matching $\mathcal{M}$ returned by the algorithm, as well as its value, which we denote by val$(\mathcal{M})$. For each $\alpha n \leq t \leq n$, define R_t as the *remaining vertices* of U when vertex v_t arrives (these are the unmatched vertices of U, after $v_1, \ldots, v_{t-1}$ are processed). With this notation, we have that

$$\text{val}(\mathcal{M}) = \sum_{t=\alpha n}^{n} \text{val}(u_t, v_t)\, \mathbf{1}_{[u_t \in R_t]}. \tag{5.3}$$

Moreover, we have the following lemma, whose proof we defer until afterwards.

Lemma 5.3. *If $f(t,n) := \alpha n/(t-1)$, then*

$$\mathbb{P}[u_t \in R_t \,|\, V_t, v_t] \geq f(t,n),$$

for $t \geq \alpha n$.

Now, $\mathrm{val}(u_t, v_t)$ and $\{u_t \in R_t\}$ are conditionally independent given (V_t, v_t), as the probes of v_t are independent from those of $v_1, \dots, v_{t-1}$. Thus,

$$\mathbb{E}[\mathrm{val}(u_t, v_t)\, \mathbf{1}_{[u_t \in R_t]} \,|\, V_t, v_t] = \mathbb{E}[\mathrm{val}(u_t, v_t) \,|\, V_t, v_t] \cdot \mathbb{P}[u_t \in R_t \,|\, V_t, v_t].$$

Moreover, for each $t \geq \alpha n$, Lemma 5.3 implies that

$$\mathbb{E}[\mathrm{val}(u_t, v_t) \,|\, V_t, v_t] \cdot \mathbb{P}[u_t \in R_t \,|\, V_t, v_t] \geq \mathbb{E}[\mathrm{val}(u_t, v_t) \,|\, V_t, v_t]\, f(t,n),$$

and so

$$\mathbb{E}[\mathrm{val}(u_t, v_t)\, \mathbf{1}_{[u_t \in R_t]} \,|\, V_t, v_t] \geq \mathbb{E}[\mathrm{val}(u_t, v_t) \,|\, V_t, v_t]\, f(t,n).$$

Thus, by applying the law of iterated expectations,

$$\begin{aligned}
\mathbb{E}[\mathrm{val}(u_t, v_t) \mathbf{1}_{[u_t \in R_t]}] &= \mathbb{E}[\, \mathbb{E}[\mathrm{val}(u_t, v_t)\, \mathbf{1}_{[u_t \in R_t]} \,|\, V_t, v_t]\,] \\
&\geq \mathbb{E}[\, \mathbb{E}[\mathrm{val}(u_t, v_t) \,|\, V_t, v_t]\, f(t,n)\,] \\
&= f(t,n)\, \mathbb{E}[\mathrm{val}(u_t, v_t)],
\end{aligned}$$

for each $t \geq \alpha n$.

As a result, using (5.3), we get that

$$\begin{aligned}
\mathbb{E}[\mathrm{val}(\mathcal{M})] &= \sum_{t=\alpha n}^{n} \mathbb{E}[\mathrm{val}(u_t, v_t)\, \mathbf{1}_{[u_t \in R_t]}] \\
&\geq \sum_{t=\alpha n}^{n} f(t,n)\, \mathbb{E}[\mathrm{val}(u_t, v_t)].
\end{aligned}$$

We may thus conclude that

$$\mathbb{E}[\mathrm{val}(\mathcal{M})] \geq \mathrm{LPOPT}_{new}(G) \sum_{t=\alpha n}^{n} \frac{f(t,n)}{n},$$

after applying (5.2).

As $\sum_{t=\alpha n}^{n} f(t,n)/n = (1 + o(1))1/e$ when $\alpha = 1/e$ (where the asymptotics are as $n \to \infty$), the result holds. □

In order to complete the proof of Theorem 5.2, we must prove Lemma 5.3. Up until now, when Algorithm 7 solves LP-new for G_t, we have been able to notate the induced edge variables as $(\widetilde{x}_{u,v})_{u \in U, v \in V_t}$ without ambiguity, despite the dependence on $\alpha n \leq t \leq n$. In the proof below, it is necessary to be more explicit in our notation, so we denote $\widetilde{x}_{u,v}$ as $\widetilde{x}_{u,v}^{(t)}$ to indicate that we are working with an edge variable from the relevant LP solution involving G_t.

Proof of Lemma 5.3. In what follows, let us assume that $\alpha n \leq t \leq n$ is fixed. We wish to prove that for each $u \in U$,

$$\mathbb{P}[u \in R_t \,|\, V_t, v_t] \geq \frac{\alpha n}{t-1}.$$

As such, we must condition on (V_t, v_t) throughout the remainder of the proof. To simplify the argument, we abuse notation slightly and remove (V_t, v_t) from the subsequent probability computations, though it is understood to implicitly appear.

Given arriving node v_j for $j = 1, \ldots, n$, once again denote $C(u, v_j)$ as the event in which v_j commits to $u \in U$. As R_t denotes the unmatched nodes after the vertices $v_1, \ldots, v_{t-1}$ are processed by Algorithm 7, observe that $u \in R_t$ if and only if $\neg C(u, v_j)$ occurs for each $j = 1, \ldots, t-1$. As a result,

$$\mathbb{P}[u \in R_t] = \mathbb{P}[\cap_{j=1}^{t-1} \neg C(u, v_j)].$$

We therefore focus on lower bounding $\mathbb{P}[\cap_{j=1}^{t-1} \neg C(u, v_j)]$ in order to prove the lemma.

First observe that for $j = 1, \ldots, \alpha n - 1$, the algorithm passes on all the trials of v_j by definition. As such, we may focus on lower bounding

$$\mathbb{P}[\cap_{j=\alpha n}^{t-1} \neg C(u, v_j)],$$

which depends only on the vertices of $V_{t-1} \setminus V_{\alpha n - 1}$. We denote $\bar{t} := t - \alpha n$ as the number of vertices within this set.

Let us first consider the vertex v_{t-1}, and the induced edge variable $\widetilde{x}_{u,v}^{(t-1)}$ for each $v \in V_{t-1}$. Observe that after applying Lemma 2.3,

$$\begin{aligned}\mathbb{P}[C(u, v_{t-1})] &= \sum_{v \in V_{t-1}} \mathbb{P}[C(u, v_{t-1}) \,|\, v_{t-1} = v] \cdot \mathbb{P}[v_{t-1} = v] \\ &= \frac{1}{t-1} \sum_{v \in V_{t-1}} \widetilde{x}_{u,v}^{(t-1)}\, p_{u,v},\end{aligned}$$

as once we condition on (V_t, v_t), v_{t-1} is uniformly distributed amongst V_{t-1}. On the other hand, the values $(\widetilde{x}_{u,v}^{(t-1)})_{u \in U, v \in V_{t-1}}$ are derived from a solution to LP-new for G_{t-1}, and so

$$\sum_{v \in V_{t-1}} \widetilde{x}_{u,v}^{(t-1)}\, p_{u,v} \leq 1.$$

We therefore get that

$$\mathbb{P}[C(u, v_{t-1})] \leq \frac{1}{t-1}.$$

Similarly, if we fix $1 \leq k \leq \bar{t}$, then we can generalize the above argument by conditioning on the identities of all the vertices preceding v_{t-k}, as well as the probes they make; that is, $(u_{t-1}, v_{t-1}), \ldots, (u_{t-(k-1)}, v_{t-(k-1)})$ (in addition to V_t and v_t as always).

In order to simplify the resulting indices, let us reorder the vertices of $V_{t-1} \setminus V_{\alpha n - 1}$. Specifically, define $\bar{v}_k := v_{t-k}, \bar{u}_k := u_{t-k}$ and $\bar{e}_k := e_{t-k}$ for $k = 1, \ldots, \bar{t}$. With this notation, we denote $\mathcal{H}_k$ as encoding the information available based on the vertices $\bar{v}_1, \ldots, \bar{v}_k$ and the edges they (potentially) committed to, namely $\bar{e}_1, \ldots, \bar{e}_k$[16]. By convention, we define $\mathcal{H}_0$ as encoding the information regarding V_t and v_t.

An analogous computation to the above case then implies that

$$\mathbb{P}[C(u, \bar{v}_k) \,|\, \mathcal{H}_{k-1}] = \sum_{v \in V_{t-k}} \widetilde{x}_{u,v}^{(t-k)}\, p_{u,v}\, \mathbb{P}[\bar{v}_k = v] \leq \frac{1}{t-k},$$

for each $k = 1, \ldots, \bar{t}$, where $\widetilde{x}_{u,v}^{(t-k)}$ is the edge variable for $v \in V_{t-k}$.

[16] Formally, $\mathcal{H}_k$ is the sigma-algebra generated from V_t, v_t and $\bar{e}_1, \ldots, \bar{e}_k$.

Observe now that in each step, we condition on strictly more information; that is, $\mathcal{H}_{k-1} \subseteq \mathcal{H}_k$ for each $k = 2, \ldots, \bar{t}$. On the other hand, observe that if we condition on $\mathcal{H}_{k-1}$ for $1 \leq k \leq \bar{t} - 1$, then the event $C(u, \bar{v}_j)$ can be determined from $\mathcal{H}_{k-1}$ for each $1 \leq j \leq k - 1$.

Using these observations, for $1 \leq k \leq \bar{t}$, the following recursion holds:

$$\begin{aligned}\mathbb{P}[\cap_{j=1}^{k} \neg C(u, \bar{v}_j)] &= \mathbb{E}\left[\mathbb{E}\left[\prod_{j=1}^{k} \mathbf{1}_{[\neg C(u,\bar{v}_j)]} \,|\, \mathcal{H}_{k-1}\right]\right] \\ &= \mathbb{E}\left[\prod_{j=1}^{k-1} \mathbf{1}_{[\neg C(u,\bar{v}_j)]}\, \mathbb{P}[\neg C(\bar{v}_k, u) \,|\, \mathcal{H}_{k-1}]\right] \\ &\geq \left(1 - \frac{1}{t-k}\right) \mathbb{P}[\cap_{j=1}^{k-1} \neg C(u, \bar{v}_j)]\end{aligned}$$

It follows that if $k = t - \alpha n$, then applying the above recursion implies that

$$\mathbb{P}[\cap_{j=\alpha n}^{t-1} \neg C(u, v_j)] \geq \prod_{k=1}^{t-\alpha n} \left(1 - \frac{1}{t-k}\right).$$

Thus, after cancelling the pairwise products,

$$\mathbb{P}[\cap_{j=\alpha n}^{t-1} \neg C(u, v_j)] \geq \frac{\alpha n}{t-1},$$

and so

$$\mathbb{P}[u \in R_t] = \mathbb{P}[\cap_{j=\alpha n}^{t-1} \neg C(u, v_j)] \geq \frac{\alpha n}{t-1}$$

thereby completing the argument.

□

6 Conclusion and Open Problems

We discussed the online stochastic matching problem in various settings and gave new and improved results with respect to a new LP relaxation which upper bounds the performance of the committal benchmark. We use our LP to create fractional solutions which can then be rounded to determine a non-adaptive sequence of edge probes. Our LP has a better stochasticity gap, as compared to the linear programs discussed in previous papers. We considered the ROM input model in the unknown stochastic graph setting, and adversarial, ROM and i.i.d. input models in the known stochastic graph setting. All of our results hold for arbitrary patience values and we consider both offline vertex weights and the more general edge weights in determining the stochastic reward.

Our results leave unsettled many interesting questions. We can view many open problems in terms of one basic issue: When (if ever) is there a provable difference between the classical online bipartite matching problem and the corresponding stochastic matching problem? What negative (i.e., inapproximation) results (if any) can be strengthened beyond what is known in the corresponding classical settings?

One of the questions we have left open is whether our competitive ratios can be seen to hold against the non-committal benchmark, or whether we must allow them more power. For instance, if our online probing algorithms execute without needing to respect commitment, is it clear that a competitive ratio of $1 - 1/e$ is attainable against the non-committal benchmark? What if we enforce commitment, but allow our algorithms to execute adaptively? We believe that these open questions highlight the difficulty of having to design probing algorithms which work for arbitrary patience constraints.

Another direction is to improve the linear program for the case of unit patience since a stochasticity gap of $1 - 1/e$ holds here, and our linear program is equivalent to those linear programs discussed in earlier papers, when restricted to unit patience. This seems to be a bottleneck in proving positive results in any model. It would also be interesting to look at other methods to prove competitive ratios without using linear programs at all (i.e., by combinatorial methods). Or when (if ever) is $1 - 1/e$ an optimal competitive ratio?

We are also interested in whether our results for stochastic matching can be extended so that offline, as well as online vertices, have patience constraints. Another extension would be to generalize the patience constraints so that now online vertices have budgets, and edges have non-uniform probing costs. The constraint would be that the cost of probes adjacent to an online vertex is limited to its budget. And finally, we are interested in whether we can obtain improved competitive ratios, for special cases, such as when the edge probabilities are decomposable or vanishingly small as studied in Goyal and Udwani [24].

Acknowledgements
We would like to thank Denis Pankratov for his helpful comments.

References

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

[2] Marek Adamczyk, Fabrizio Grandoni, Stefano Leonardi, and Michal Wlodarczyk. When the optimum is also blind: a new perspective on universal optimization. In *ICALP*, 2017.

[3] Marek Adamczyk, Fabrizio Grandoni, and Joydeep Mukherjee. Improved approximation algorithms for stochastic matching. In Nikhil Bansal and Irene Finocchi, editors, *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2015.

[4] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1253–1264, 2011.

[5] Arash Asadpour and Hamid Nazerzadeh. Maximizing stochastic monotone submodular functions. *Management Science*, 62(8):2374–2391, 2016.

[6] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.

[7] Alok Baveja, Amit Chavan, Andrei Nikiforov, Aravind Srinivasan, and Pan Xu. Improved bounds in stochastic matching and optimization. *Algorithmica*, 80(11):3225–3252, Nov 2018.

[8] Allan Borodin, Calum MacRury, and Akash Rakheja. Greedy approaches to online stochastic matching, 2020.

[9] Brian Brubach, Nathaniel Grammel, and Aravind Srinivasan. Vertex-weighted online stochastic matching with patience constraints. *CoRR*, abs/1907.03963, 2019.

[10] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. New algorithms, better bounds, and a novel model for online stochastic matching. In *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*, pages 24:1–24:16, 2016.

[11] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Attenuate locally, win globally: Attenuation-based frameworks for online stochastic matching with timeouts. *Algorithmica*, 82(1):64–87, 2020.

[12] Ning Chen, Nicole Immorlica, Anna R. Karlin, Mohammad Mahdian, and Atri Rudra. Approximating matches made in heaven. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I*, ICALP '09, pages 266–278, 2009.

[13] Kevin P. Costello, Prasad Tetali, and Pushkar Tripathi. Stochastic matching with commitment. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming*, pages 822–833, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[14] Brian C. Dean, Michel X. Goemans, and Jan Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 395–404, 2005.

[15] Brian C. Dean, Michel X. Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Math. Oper. Res.*, 33(4):945–964, 2008.

[16] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 101–107, Philadelphia, PA, USA, 2013. Society for Industrial and Applied Mathematics.

[17] Soheil Ehsani, MohammadTaghi Hajiaghayi, Thomas Kesselheim, and Sahil Singla. Prophet secretary for combinatorial auctions and matroids. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, page 700–714, USA, 2018. Society for Industrial and Applied Mathematics.

[18] Hossein Esfandiari, Mohammad Taghi Hajiaghayi, Brendan Lucier, and Michael Mitzenmacher. Online pandora's boxes and bandits. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 1885–1892, 2019.

[19] Jon Feldman, Aranyak Mehta, Vahab S. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 117–126, 2009.

[20] Hao Fu, Jian Li, and Pan Xu. A PTAS for a Class of Stochastic Dynamic Programs. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 56:1–56:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[21] Buddhima Gamlath, Sagar Kale, and Ola Svensson. Beating greedy for stochastic bipartite matching. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, page 2841–2854, USA, 2019. Society for Industrial and Applied Mathematics.

[22] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *J. ACM*, 53(3):324–360, May 2006.

[23] Bernd Gärtner and Jirí Matousek. *Understanding and using linear programming.* Universitext. Springer, 2007.

[24] Vineet Goyal and Rajan Udwani. Online matching with stochastic rewards: Optimal competitive ratio via path based formulation. In Péter Biró, Jason Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, page 791. ACM, 2020.

[25] Zhiyi Huang, Zhihao Tang, Xiaowei Wu, and Yuhao Zhang. Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals. *ACM Transactions on Algorithms*, 15, 04 2018.

[26] Zhiyi Huang and Qiankun Zhang. Online primal dual meets online matching with stochastic rewards: Configuration lp to the rescue. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 1153–1164, New York, NY, USA, 2020. Association for Computing Machinery.

[27] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 587–596, 2011.

[28] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 352–358, 1990.

[29] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms – ESA 2013*, pages 589–600, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[30] Euiwoong Lee and Sahil Singla. Optimal Online Contention Resolution Schemes via Ex-Ante Prophet Inequalities. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 57:1–57:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[31] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: An approach based on strongly factor-revealing lps. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 597–606, New York, NY, USA, 2011. ACM.

[32] Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. *Math. Oper. Res.*, 37(4):559–573, 2012.

[33] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[34] Aranyak Mehta and Debmalya Panigrahi. Online matching with stochastic rewards. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 728–737. IEEE Computer Society, 2012.

[35] D. Seese. Groetschel, m., l. lovasz, a. schrijver: Geometric algorithms and combinatorial optimization. (algorithms and combinatorics. eds.: R. l. graham, b. korte, l. lovasz. vol. 2), springer-verlag 1988, xii, 362 pp., 23 figs., dm 148,-. isbn 3–540–13624-x. *Biometrical Journal*, 32(8):930–930, 1990.

[36] Danny Segev and Sahil Singla. Efficient approximation schemes for stochastic probing and prophet problems, 2020.

[37] Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. A simple 1-1/e approximation for oblivious bipartite matching, 2020.

[38] Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. Towards a better understanding of randomized greedy matching. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1097–1110, 2020.

[39] Martin Weitzman. Optimal search for the best alternative. *Econometrica*, 47:641–654, 1979.

[40] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, USA, 1st edition, 2011.

A Relaxing the Committal Benchmark

In this section, we consider the committal benchmark, as defined in Section 2. In particular, we prove that LP-new is a relaxation of the committal benchmark (Theorem 2.2). It is convenient to extend our definition of an online probing algorithm to the offline setting (as has previously been implicitly suggested by the committal and non-committal benchmarks).

Suppose that we are given an arbitrary stochastic graph $G = (U, V, E)$. We define an **(offline) probing algorithm** as an algorithm which adaptively reveals the edge states G, while respecting the patience values of G. Notably, we do *not* restrict a probing algorithm to any specific ordering of the edges of G. The goal of a probing algorithm is again to return a matching of active edges of large expected weight, though it must **respect commitment**. That is, it has the property that if it makes a probe which yields an active edge, then this edge must be included in the current matching (if possible). The value of the committal benchmark on G, denoted $\text{OPT}(G)$, simply corresponds to the largest expected value a probing algorithm can attain on G.

Let us now restate LP-new for convenience:

$$\text{maximize} \quad \sum_{v \in V} \sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} \left(\sum_{i=1}^{|\boldsymbol{u}|} w_{u_i,v} \, g_v^i(\boldsymbol{u}) \right) \cdot x_v(\boldsymbol{u}) \tag{LP-new}$$

$$\text{subject to} \quad \sum_{v \in V} \sum_{i=1}^{\ell_v} \sum_{\substack{\boldsymbol{u}^* \in U^{(\leq \ell_v)}: \\ u_i^* = u}} g_v^i(\boldsymbol{u}^*) \cdot x_v(\boldsymbol{u}^*) \leq 1 \qquad \forall u \in U \tag{A.1}$$

$$\sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} x_v(\boldsymbol{u}) \leq 1 \qquad \forall v \in V, \tag{A.2}$$

$$x_v(\boldsymbol{u}) \geq 0 \qquad \forall v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)} \tag{A.3}$$

In order to prove that LP-new is a relaxation of the committal benchmark, we must show that $\text{OPT}(G) \leq \text{LPOPT}_{new}(G)$. Observe that in the above terminology, this is equivalent to showing that for each (offline) probing algorithm $\mathcal{A}$, $\mathbb{E}[\text{val}(\mathcal{A}(G))] \leq \text{LPOPT}_{new}(G)$, where $\mathcal{A}(G)$ is the matching returned by $\mathcal{A}$.

Suppose now that we define $x_v(\boldsymbol{u})$ to be the probability that $\mathcal{A}$ probes the edges $(u_i, v)_{i=1}^{|\boldsymbol{u}|}$ in order, where $v \in V$ and $\boldsymbol{u} \in U^{(\leq \ell_v)}$. Let us suppose that $\mathcal{A}$ has the following property:

1. For each $v \in V$, the edge probes involving v are made independently of the edge states $(\text{st}(u, v))_{u \in U}$.

Observe that by using (1), the expected value of the edge assigned to v is

$$\sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} \left(\sum_{i=1}^{|\boldsymbol{u}|} w_{u_i, v} \cdot g_v^i(\boldsymbol{u}) \right) \cdot x_v(\boldsymbol{u}).$$

By additionally arguing that $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v}}$ is a feasible solution to LP-new, we can then upper bound $\mathbb{E}[\text{val}(\mathcal{A}(G)]$ by $\text{LPOPT}_{new}(G)$.

That being said, since we make no assumption on how $\mathcal{A}$ moves between edge probes, it is not clear that we can assume it satisfies (1) without loss of generality. As such, the natural interpretation of the variables of LP-new does not seem to easily lend itself to a proof of Theorem 2.2.

In order to get around these issues, we introduce a new stochastic probing problem for a stochastic graph $G = (U, V, E)$, known as the **relaxed stochastic matching problem**. This problem is closely related to the stochastic matching problem, however it places fewer restrictions on how many times each vertex $u \in U$ may be matched by a probing algorithm. As such, it has an optimum solution which upper bounds $\text{OPT}(G)$. Interestingly, LP-new exactly encodes this relaxed matching problem, which implies Theorem 2.2 as a corollary.

We now introduce the definition of a **relaxed probing algorithm**, described in the following way:

A relaxed probing algorithm probes edges of G, while respecting the patience constraints of the online nodes of V. It is allowed to arbitrarily move between the edges of G, and it must return $\mathcal{M}$, a subset of its probes which yielded active edges. The goal of the relaxed probing algorithm is to maximize the expected weight of $\mathcal{M}$, while ensuring that the following properties are satisfied:

1. Each $v \in V$ appears in at most one edge of $\mathcal{M}$.

2. For each $u \in U$, the *expected* number of edges which contain u is at most one.

We refer to $\mathcal{M}$ as a **one-sided matching** for the online nodes. In a slight abuse of terminology, we say that a relaxed probing algorithm **matches** the edge e, provided e is included in $\mathcal{M}$.

A relaxed probing algorithm must **respect commitment**. That is, it has the property that if a probe to $e = (u, v)$ yields an active edge, then the edge is included in $\mathcal{M}$ (provided v is currently not in $\mathcal{M}$). Observe that this requires the relaxed probing algorithm to include e, even if u is already adjacent to some element of $\mathcal{M}$.

We define the **relaxed benchmark** as the optimum relaxed probing algorithm on G, and denote $\mathrm{OPT}_{rel}(G)$ as the value this benchmark attains G. Observe that by definition,

$$\mathrm{OPT}(G) \leq \mathrm{OPT}_{rel}(G),$$

where $\mathrm{OPT}(G)$ is the value of the committal benchmark on G.

Finally, we say that a relaxed probing algorithm is **non-adaptive**, provided its edge probes are statistically independent from the edge states of G; that is, the random variables $(\mathrm{st}(e))_{e \in E}$. We emphasize that this is equivalent to specifying an ordering λ on a subset of E, where each vertex $v \in V$ appears in at most ℓ_v edges of λ. Once λ is generated (potentially using randomness), the edges specified by λ are probed in order, and an edge is added to the matching, provided its online node is unmatched.

Unlike the committal benchmark, $\mathrm{OPT}_{rel}(G)$ can be attained by a non-adaptive relaxed probing algorithm.

Theorem A.1. *There exists a relaxed probing algorithm which is non-adaptive and attains value $OPT_{rel}(G)$ in expectation.*

We defer the proof of Theorem A.1 for now, and instead show how it allows us to prove Theorem 2.2. In fact, we prove that LP-new encodes the value of the relaxed benchmark exactly, thus implying Theorem 2.2 since $\mathrm{OPT}(G) \leq \mathrm{OPT}_{rel}(G)$.

Theorem A.2. *For any stochastic graph G, an optimum solution to LP-new has value equal to $OPT_{rel}(G)$, the value of the relaxed benchmark on G.*

Proof. Suppose we are presented a solution $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$ to LP-new. We observe then the following relaxed probing algorithm:

1. $\mathcal{M} \leftarrow \emptyset$.
2. For each $v \in V$, set $e \leftarrow \textsc{VertexProbe}(G, (x_v(\boldsymbol{u}))_{\boldsymbol{u} \in U^{(\leq \ell_v)}}, v)$.
3. If $e \neq \emptyset$, then let $e = (u, v)$ and set $\mathcal{M}(v) = u$.
4. Return $\mathcal{M}$.

Using Lemma 2.3, it is clear that

$$\mathbb{E}[\mathrm{val}(\mathcal{M})] = \sum_{v \in V} \sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} \left(\sum_{i=1}^{|\boldsymbol{u}|} w_{u_i, v} \, g_v^i(\boldsymbol{u}) \right) \cdot x_v(\boldsymbol{u}).$$

Moreover, each vertex $u \in U$ is matched by $\mathcal{M}$ at most once in expectation, as a consequence of (A.1).

In order to complete the proof, it remains to show that if $\mathcal{A}$ is an optimum relaxed probing algorithm, then there exists a solution to LP-new whose value is equal to $\mathbb{E}[\mathrm{val}(\mathcal{A}(G))]$ (where $\mathcal{A}(G)$ is the one-sided matching returned by $\mathcal{A}$). In fact, by Theorem A.1, we may assume that $\mathcal{A}$ is non-adaptive.

Observe then that for each $v \in V$ and $\boldsymbol{u} = (u_1, \ldots, u_k) \in U^{(\leq \ell_v)}$ with $k = |\boldsymbol{u}|$ we can define

$$x_v(\boldsymbol{u}) := \mathbb{P}[\mathcal{A} \text{ probes the edges } (u_i, v)_{i=1}^k \text{ in order}].$$

Setting $\mathcal{M} = \mathcal{A}(G)$ for convenience, observe that if $\text{val}(\mathcal{M}(v))$ corresponds to the weight of the edge assigned to v (which is 0 if no assignment is made), then

$$\mathbb{E}[\text{val}(\mathcal{M}(v))] = \sum_{\boldsymbol{u} \in U^{(\leq \ell_v)}} \left(\sum_{i=1}^{|\boldsymbol{u}|} w_{u_i,v} \, g_v^i(\boldsymbol{u}) \right) \cdot x_v(\boldsymbol{u}),$$

as $\mathcal{A}$ is non-adaptive.

Moreover, for each $u \in U$,

$$\sum_{v \in V} \sum_{i=1}^{\ell_v} \sum_{\substack{\boldsymbol{u}^* \in U^{(\leq \ell_v)}: \\ u_i^* = u}} g_v^i(\boldsymbol{u}^*) \cdot x_v(\boldsymbol{u}^*) \leq 1,$$

by once again using the non-adaptivity of $\mathcal{A}$. The proof is therefore complete.

□

A.1 Non-adaptivity in the Relaxed Stochastic Matching Problem

We now argue that in the relaxed stochastic matching problem, the relaxed benchmark does not require adaptivity. Our approach is to first generalize the LP used by Gamlath et al. [21] to the case of arbitrary patience. In doing so, we argue that this generalized LP also encodes the relaxed stochastic matching problem. Moreover, given an optimum solution to this LP, we can employ the techniques of Gamlath et al. [21] and Costello et al. [13] to recover an optimum probing algorithm which is non-adaptive. These observations immediately imply Theorem A.1, which allows us to complete the proof of Theorem 2.2.

Suppose that $G = (U, V, E)$ is an arbitrary stochastic graph. For each $S \subseteq U$ and $v \in V$, we first define

$$p(S, v) := 1 - \prod_{u \in S} (1 - p_{u,v}),$$

which corresponds to the probability that an edge between v and S is active. Moreover, for $1 \leq k \leq |U|$, denote $\binom{U}{k}$ as the collection of subsets of U of size k, and $\binom{U}{\leq k} = \cup_{i=1}^{k} \binom{U}{i}$ as the collection of (non-empty) subsets of U of size no greater than k.

For each $v \in V$ and $R \in \binom{U}{\leq \ell_v}$, we interpret the variable $\alpha_v(R)$ as the probability that the relaxed benchmark probes the edges $\{v\} \times R$. Moreover, for $u \in U$, we interpret the variable $z_{u,v}(R)$ as corresponding to the probability that the relaxed probing algorithm probes the edges $\{v\} \times R$ *and* matches the edge (u, v). In this way, the edge probability $p_{u,v}$ is implicitly encoded in this variable.

$$\text{maximize} \quad \sum_{u \in U, v \in V} \sum_{R \in \binom{U}{\leq \ell_v}} w_{u,v} \cdot z_{u,v}(R) \tag{LP-rel}$$

$$\text{subject to} \quad \sum_{v \in V} \sum_{R \in \binom{U}{\leq \ell_v}} z_{u,v}(R) \leq 1 \qquad \forall u \in U \tag{A.4}$$

$$\sum_{u \in S} z_{u,v}(R) \leq p(S, v) \cdot \alpha_v(R) \quad \forall v \in V,\, R \in \binom{U}{\leq \ell_v}, S \subseteq R \tag{A.5}$$

$$\sum_{R \in \binom{U}{\leq \ell_v}} \alpha_v(R) \leq 1 \qquad \forall v \in V \tag{A.6}$$

$$z_{u,v}(R) = 0 \qquad \forall u \in U \setminus R, v \in V, R \in \binom{U}{\leq \ell_v} \tag{A.7}$$

$$z_{u,v}(R) \geq 0 \qquad \forall u \in U, v \in V, R \in \binom{U}{\leq \ell_v} \tag{A.8}$$

$$\alpha_v(R) \geq 0 \qquad \forall v \in V, R \in \binom{U}{\leq \ell_v} \tag{A.9}$$

Remark. *When $G = (U, V, E)$ has full patience, $\alpha_v(U)$ may be set to 1 for each $v \in V$. As a result, only the $z_{u,v}(U)$ variables become relevant for $u \in U$, and so LP-rel generalizes the LP of Gamlath et al. in [21] (see (2.7) of Section 2).*

If $\text{LPOPT}_{rel}(G)$ corresponds to the optimum value of this LP, then this value encodes $\text{OPT}_{rel}(G)$ exactly. Moreover, an optimum solution to LP-rel induces a relaxed probing algorithm which is non-adaptive:

Theorem A.3. *For any stochastic graph G, we have that*

$$LPOPT_{rel}(G) = OPT_{rel}(G).$$

Moreover, there exists a relaxed probing algorithm which is non-adaptive and optimum, thereby proving Theorem A.1.

Before proving this theorem, we first state a key result from the work of Gamlath et al. [21], which helps motivate constraint (A.5) of LP-rel. We mention that an almost identical guarantee is also proven by Costello et al. in [13].

Theorem A.4 ([21])**.** *Suppose that $G = (U, V, E)$ is a stochastic graph, and $v \in V$ and $R \subseteq U$ are fixed. Assume that there are non-negative values $(y_{u,v})_{u \in R}$, such that*

$$\sum_{u \in S} y_{u,v} \leq p(v, S) \tag{A.10}$$

for each $S \subseteq R$. Under these assumptions, there exists a non-adaptive and committal probing algorithm, say $\mathcal{B}_v(R)$, which processes the single online node v while executing on the stochastic sub-graph $G[\{v\} \cup R]$. Moreover, it has the guarantee that

$$\mathbb{P}[\mathcal{B}_v(R) \text{ matches } v \text{ to } u] = y_{u,v}$$

for each $u \in R$.

Proof of Theorem A.3. We first argue that the relaxed benchmark corresponds to a solution of LP-rel. Observe that since the relaxed benchmark respects commitment, we may assume that for each $v \in V$, if a probe involving v yields an active edge, then v is never probed again.

For each $v \in V$ and $R \subseteq U$, where $k := |R|$ satisfies $k \leq \ell_v$, define $\alpha_v(R)$ as the probability that the relaxed benchmark probes the k edges between R and v.

If we now take $u \in U$, then define $z_{u,v}(R)$ as the probability that the relaxed benchmark probes the edges $R \times \{v\}$, *and* matches u to v. For convenience, we also define $z_{u,v}$ as the probability that the relaxed benchmark matches u to v. We then get that

$$z_{u,v} = \sum_{R \in \binom{U}{\leq \ell_v}} z_{u,v}(R)$$

for each $u \in U$ and $v \in V$. If $\mathcal{M}$ corresponds to the one-sided matching returned by the relaxed benchmark, then observe that

$$\mathbb{E}[\text{val}(\mathcal{M})] = \sum_{u \in U, v \in V} w_{u,v} \cdot z_{u,v}. \tag{A.11}$$

We now claim that $(z_{u,v}(R))_{u \in U, v \in V, R \in \binom{U}{\leq \ell_v}}$ together with $(\alpha_v(R))_{v \in V, R \in \binom{U}{\leq \ell_v}}$ corresponds to a feasible solution to LP-rel. Given $v \in V$, $R \subseteq U$, and $S \subseteq R$, we focus on proving that (A.5) holds, as the other constraints are easily seen to hold. For notational simplicity, we focus on the case of this constraint when $S = R$, however the general case follows identically.

Let us define $E_v(R)$ as the event in which the relaxed benchmark probes the edges $\{v\} \times R$. Observe now that if the relaxed benchmark matches v to some vertex of R, then one of the edges of $\{v\} \times R$ must have been active. As a result,

$$\sum_{u \in R} z_{u,v}(R) \leq \mathbb{P}[E_v(R) \cap \bigcup_{u \in R} \{\text{st}(u,v) = 1\}]. \tag{A.12}$$

On the other hand, we claim that the events $E_v(R)$ and $\cup_{u \in R}\{\text{st}(u,v) = 1\}$ are negatively correlated. That is,

$$\mathbb{P}[E_v(R) \cap \bigcup_{u \in R} \{\text{st}(u,v) = 1\}] \leq \mathbb{P}[E_v(R)] \cdot \mathbb{P}[\cup_{u \in R}\{\text{st}(u,v) = 1\}] = \alpha_v(R) \cdot p(v,R), \tag{A.13}$$

where $p(v,R) = 1 - \prod_{u \in R}(1 - p_{u,v})$. To see this, for each $i \in [\ell_v]$ define X_v^i as the offline vertex of the i^{th} edge involving v which is probed by the relaxed benchmark. By convention, if no such vertex exists, then $X_v^i := \emptyset$. Set $k = |R|$, and define $R^{(k)}$ as the set of k-length tuples of R whose coordinates are all distinct. Observe that since the relaxed benchmark stops probing edges adjacent to v as soon as it witnesses an active edge involving v, $E_v(R) \cap \bigcup_{u \in R}\{\text{st}(u,v) = 1\}$ occurs if and only if there exists some $\boldsymbol{u} \in R^{(k)}$ such that

$$\{\text{st}(u_k, v) = 1 \text{ and } X_v^k = u_k\} \cap \bigcap_{i=1}^{k-1} \{X_v^i = u_i \text{ and } \text{st}(u_i, v) = 0\}.$$

Moreover, for any $\boldsymbol{u} \in R^{(k)}$, the relaxed benchmark must decide whether to probe (u_k, v) before observing $\text{st}(u_k, v)$. Thus,

$$\mathbb{P}[\{\text{st}(u_k, v) = 1 \text{ and } X_v^k = u_k\} \cap \bigcap_{i=1}^{k-1} \{X_v^i = u_i \text{ and } \text{st}(u_i, v) = 0\}]$$

is equal to

$$p_{u_k,v} \cdot \mathbb{P}[\{X_v^k = u_k \cap \bigcap_{i=1}^{k-1} \{X_v^i = u_i \text{ and } \text{st}(u_i, v) = 0\}],$$

which itself is upper bounded by

$$p(v,R) \cdot \mathbb{P}[\{X_v^k = u_k\} \cap \bigcap_{i=1}^{k-1} \{X_v^i = u_i \text{ and } \text{st}(u_i, v) = 0\}].$$

Thus, (A.13) holds after summing over all $\boldsymbol{u} \in R^{(k)}$. As such, combined with (A.12), it follows that constraint (A.5) must be satisfied. Since the remaining constraints of LP-rel, are easily seen to hold, we may apply (A.11), to conclude that $\text{OPT}_{rel}(G) \leq \text{LPOPT}_{rel}(G)$.

Let us now suppose that we are presented a solution to LP-rel, which we denote by $(z_{u,v}(R))_{u \in U, v \in V, R \in \binom{U}{\leq \ell_v}}$ and $(\alpha_v(R))_{v \in V, R \in \binom{U}{\leq \ell_v}}$. Using this solution, we can derive relaxed probing algorithm which returns a one-sided matching whose expected value is equal to the LP solution's value.

Let us first fix a vertex $v \in V$ and consider the values $(\alpha_v(R))_{R \in \binom{U}{\leq \ell_v}}$ and $(z_{u,v}(R))_{u \in U, R \in \binom{U}{\leq \ell_v}}$, where $\alpha_v(R) \neq 0$. Observe that if we fix $R \subseteq U$, $1 \leq |R| \leq \ell_v$, then the values $(z_{u,v}(R)/\alpha_v(R))_{u \in R}$ satisfy the relevant inequalities of (A.10) thanks to constraint (A.5) of LP-rel. Theorem A.4 thus guarantees that there exists a committal probing algorithm for $G[R \cup \{v\}]$, say $\mathcal{B}_v(R)$, such that

$$\mathbb{P}[\mathcal{B}_v(R) \text{ matches } v \text{ to } u] = \frac{z_{u,v}(R)}{\alpha_v(R)}$$

for each $u \in U$. Moreover, $\mathcal{B}_v(R)$ is non-adaptive; that is, the probes of $\mathcal{B}_v(R)$ are statistically independent from the edge states, $(\mathrm{st}(u,v))_{u \in R}$.

This suggests the following relaxed probing algorithm, which we denote by $\mathcal{B}$:

1. Set $\mathcal{M} = \emptyset$.
2. For each $v \in V$, pass on v with probability $1 - \sum_{R \in \binom{U}{\leq \ell_v}} \alpha_v(R)$
3. Otherwise, draw $P \subseteq U$ with probability $\alpha_v(P)$.
4. Execute $\mathcal{B}_v(P)$, and match v to whichever vertex of U (if any) v is matched to by $\mathcal{B}_v(P)$.

Observe now that

$$\mathbb{P}[\mathcal{B} \text{ matches } v \text{ to } u] = \sum_{R \in \binom{U}{\leq \ell_v}} \alpha_v(R) \cdot \frac{z_{u,v}(R)}{\alpha_v(R)} = z_{u,v},$$

Thus,

$$\mathbb{E}[\mathrm{val}(\mathcal{M})] = \sum_{u \in U, v \in V} w_{u,v} \cdot z_{u,v}.$$

Moreover, if N_u counts the number of vertices of V which match to u, then

$$\mathbb{E}[N_u] = \sum_{v \in V} z_{u,v} \leq 1,$$

as the values $(z_{u,v})_{v \in V}$ satisfy (A.4) by assumption.

Finally, we observe that $\mathcal{B}$ respects commitment and is non-adaptive.

□

We remark that LP-rel, coupled with the probing procedure of Theorem A.4, can be used to construct online probing algorithms with the same competitive guarantees as derived using LP-new and VertexProbe. That being said, LP-rel does not seem to be poly-time solvable, at least for arbitrary patience[17]. Introducing LP-new allows us to derive online probing algorithms which are poly-time solvable and which do not require appealing to the subroutine involved in Theorem A.4

[17] If $\max_{v \in V} \ell_v$ is upper bounded by a constant, independent of the size of U, then the LP is solvable using the separation oracle presented by Gamlath et al. [21]. A similar statement is true if all the patience values are close to $|U|$.

B Solving LP-new Efficiently

In this section, we prove the following result:

Theorem B.1. *An optimum solution to LP-new can be found in polynomial time in the size of the stochastic graph, $G = (U, V, E)$.*

In order to prove this claim, it suffices to show that LP-new-dual has a (deterministic) polynomial time separation oracle, as a consequence of how the ellipsoid algorithm [35, 23] executes (see [40, 2, 30] for applications). As such, we restate the dual of LP-new for convenience:

$$
\begin{array}{llll}
\text{minimize} & \displaystyle\sum_{u \in U} \alpha_u + \sum_{v \in V} \beta_v & & \text{(LP-new-dual)} \\
\text{subject to} & \displaystyle\beta_v + \sum_{j=1}^{|\boldsymbol{u}^*|} g_v^j(\boldsymbol{u}^*) \cdot \alpha_{u_j^*} \geq \sum_{j=1}^{|\boldsymbol{u}^*|} w_{u_j^*,v} \cdot g_v^j(\boldsymbol{u}^*) & \forall v \in V, \boldsymbol{u}^* \in U^{(\leq \ell_v)} & \text{(B.1)} \\
& \alpha_u \geq 0 & \forall u \in U & \text{(B.2)} \\
& \beta_v \geq 0 & \forall v \in V & \text{(B.3)}
\end{array}
$$

Suppose now that we are presented a particular selection dual variables, say $((\alpha_u)_{u \in U}, (\beta_v)_{v \in V})$, which may or may not be a feasible solution to LP-new-dual. Our separation oracle must determine efficiently whether these variables satisfy all the constraints of LP-new-dual. In the case in which the solution is *infeasible*, the oracle must additionally return a constraint which is violated.

It is clear that we can accomplish this for the non-negativity constraints, so we hereby assume that $\alpha_u \geq 0$ and $\beta_v \geq 0$ for all $u \in U$ and $v \in V$.

Let us now fix a particular $v \in V$ in what follows. We wish to determine whether there exists some $\boldsymbol{u}^* \in U^{(\leq \ell_v)}$ such that

$$\beta_v + \sum_{j=1}^{|\boldsymbol{u}^*|} g_v^j(\boldsymbol{u}^*) \cdot \alpha_{u_j^*} < \sum_{j=1}^{|\boldsymbol{u}^*|} w_{u_j^*,v} \cdot g_v^j(\boldsymbol{u}^*).$$

To make such a determination, we consider the function ϕ, where

$$\phi(\boldsymbol{u}^*) := \sum_{j=1}^{|\boldsymbol{u}^*|} (w_{u_j^*,v} - \alpha_{u_j^*}) \cdot g_v^j(\boldsymbol{u}^*), \tag{B.4}$$

for $\boldsymbol{u}^* \in U^{(\leq \ell_v)}$. Our goal is to verify whether there exists some $\boldsymbol{u}^* \in U^{(\leq \ell_v)}$ such that $\phi(\boldsymbol{u}^*) > \beta_v$. If we can efficiently check this for a fixed $v \in V$, then we can iterate the same procedure for all $v \in V$, thus yielding a polynomial time separation oracle for LP-new-dual. Thus, in order to prove Theorem B.1, we only need to prove the following statement:

Proposition B.2. *There exists an efficient deterministic algorithm which checks whether there exists some $\boldsymbol{u}^* \in U^{(\leq \ell_v)}$ such that $\phi(\boldsymbol{u}^*) > \beta_v$. Moreover, if a tuple with this property exists, then this algorithm will return such a tuple in polynomial time.*

In [9], Brubach et al. consider the setting in which one is presented *non-negative* edge weights $(\bar{w}_{u^*,v})_{u^* \in U}$, and the function

$$\psi(\boldsymbol{u}^*) := \sum_{j=1}^{|\boldsymbol{u}^*|} \bar{w}_{u_j^*,v} \cdot g_v^j(\boldsymbol{u}^*), \tag{B.5}$$

for $\boldsymbol{u}^* \in U^{(\leq \ell_v)}$. They show that one can maximize this function in polynomial time using a deterministic algorithm based on dynamical programming techniques.

Theorem B.3 ([9])**.** *For any $v \in V$ with patience ℓ_v and non-negative selection of weights, $(\bar{w}_{u^*,v})_{u^* \in U}$, the function $\psi = \psi(\boldsymbol{u}^*)$ in* (B.5) *can be maximized in polynomial time using a deterministic procedure.*

We can apply Theorem B.3 to prove Proposition B.2.

Proof of Proposition B.2. Let us first define $\bar{w}_{u,v} := w_{u,v} - \alpha_u$ for all $u \in U$. Denote P as those $u \in U$ such that $\bar{w}_{u,v} \geq 0$. First note that if P is empty, then clearly $\phi(\boldsymbol{u}^*) \leq 0$ for all $\boldsymbol{u} \in U^{(\leq \ell_v)}$, so since $\beta_v \geq 0$ by assumption, there is nothing to prove.

Let us therefore assume that $P \neq \emptyset$. Observe then that we can restrict our attention to those $\boldsymbol{u}^* \in U^{(\leq \ell_v)}$ whose entries all lie in P; namely, $P^{(\leq \ell_v)}$. By applying Theorem B.3, we are guaranteed a deterministic procedure for maximizing ϕ on $P^{(\leq \ell_v)}$ in polynomial time. Let us denote the outcome of this procedure by $\boldsymbol{u}_{max}$. Observe then that either

$$\beta_v \geq \phi(\boldsymbol{u}_{max}) \geq \phi(\boldsymbol{u}^*)$$

for all $\boldsymbol{u}^* \in U^{(\leq \ell_v)}$, or $\boldsymbol{u}_{max}$ satisfies

$$\phi(\boldsymbol{u}_{max}) > \beta_v.$$

In either case, the procedure satisfies the requirements of Proposition B.2, and so the proof is complete.

□

We conclude the section by considering the stochastic known i.i.d. setting. Specifically, consider a stochastic type graph, say $G = (U, V, E)$, with n arrivals draw from the fractional rates, $\boldsymbol{r} = (r_v)_{v \in V}$. In this case, we can solve LP-new-iid efficiently by presenting a separation oracle for its dual. The reduces to the same maximization problem just considered, with the caveat that for a fixed type node $v \in V$, we compare the maximized value to $\beta_v \cdot r_v$ (where β_v is the dual variable associated to $v \in V$).

C The Non-committal Benchmark

In this section, we extend our definition of a probing algorithm to general (i.e., not necessarily bipartite) stochastic graphs, as well as our definitions of the committal and non-committal benchmarks. We then review LP-std, and show that it is in fact a relaxation of the non-committal benchmark. This allows us to also prove that LP-std-iid is a relaxation of the non-committal benchmark, as defined for the stochastic known i.i.d. matching problem. As a corollary, we argue that many of the results in the stochastic matching literature hold against this stronger benchmark.

We then discuss LP-new, and discuss the restricted settings in which is it also a relaxation of the non-committal benchmark (when once again restricted to online stochastic matching problems). In the case of arbitrary patience values, we show that LP-new is *not* a relaxation of the non-committal benchmark, and discuss the limitations of our techniques. In particular, we explain the difficulty in designing probing algorithms which respect commitment and also attain large competitive ratios against the non-committal benchmark.

C.1 Relaxing the Non-committal Benchmark via LP-std

Let us suppose that $G = (V, E)$ is a general stochastic graph with edge weights $(w_e)_{e \in E}$, edge probabilities $(p_e)_{e \in E}$ and patience values $(\ell_v)_{v \in V}$. We emphasize that G need not be bipartite.

An **(offline) probing algorithm** must satisfy the requirement that most ℓ_v probes are made to the neighbouring edges of v for each $v \in V$. We say that a probing algorithm is **committal**, provided it satisfies the following property when constructing its matching: if an edge $e \in E$ is probed, then $e = (u, v)$ must be added to the current matching, provided u and v are currently unmatched. Alternatively, a probing algorithm is said to be **non-committal**, provided the matching it constructs is done in the following manner: if the algorithm (adaptively) reveals the edges $B \subseteq E$ to be active, then an optimum matching constructed from B is returned.

We can also extend our definitions of the committal and non-committal benchmarks to G. Specifically, the committal benchmark corresponds to the optimum committal probing algorithm, whose expected value we denote by $\text{OPT}(G)$. Similarly, the non-committal benchmark corresponds to the optimum non-committal probing algorithm, whose expected value we denote by $\text{OPT}_{non}(G)$.

We now generalize LP-std to the non-bipartite case, as originally presented in [6] by Bansal et al.:

$$\text{maximize} \quad \sum_{e \in E} w_e \cdot p_e \cdot x_e \tag{LP-std}$$

$$\text{subject to} \quad \sum_{\substack{e \in E: \\ v \in e}} x_e \leq \ell_v \qquad \forall v \in V \tag{C.1}$$

$$\sum_{\substack{e \in E: \\ v \in e}} p_e \cdot x_e \leq 1 \qquad \forall v \in V \tag{C.2}$$

$$\tag{C.3}$$

$$0 \leq x_e \leq 1 \qquad \forall e \in E \tag{C.4}$$

Bansal et al. showed that LP-std is a relaxation of the committal benchmark; that is, $\text{OPT}(G) \leq \text{LPOPT}_{std}(G)$, where $\text{LPOPT}_{std}(G)$ denotes the optimum value of LP-std. We claim that this LP-std is also a relaxation of the non-committal benchmark:

Theorem C.1. *For any (general) stochastic graph $G = (V, E)$, it holds that*

$$OPT_{non}(G) \leq LPOPT_{std}(G).$$

As a result, if an (offline) probing algorithm attains an approximation ratio of $0 \leq c \leq 1$ against LP-std, then it attains this approximation ratio against *non-committal* benchmark as well. Notably, this implies that the competitive ratios of the probing algorithms considered in [6, 3, 7] all in fact hold against the non-committal benchmark.

We can also consider the known i.i.d. setting, in which we are presented a known i.i.d. instance $(G, \boldsymbol{r}, n)$. In this case, an analogous statement of Theorem C.1 holds regarding LP-std-iid:

Theorem C.2. *For any input $(G, \boldsymbol{r}, n)$ of the known i.i.d. stochastic matching problem,*

$$OPT_{non}(G, \boldsymbol{r}, n) \leq LPOPT_{std-iid}(G, \boldsymbol{r}, n).$$

Since the works of [6, 3, 10, 11] all prove competitive ratios against LP-std-iid, Theorem C.2 implies that these competitive ratios all hold against the non-committal benchmark.

In the remainder of the section, we prove Theorem C.1, which we then argue can be used to imply Theorem C.2. It will be convenient to instead work with a modified version of LP-std, where each edge $e \in E$ is instead associated with two variables, namely x_e and z_e. We interpret the former variable as the probability that the non-committal benchmark probes the edge e, whereas the latter variable corresponds to the probability that e is included in the matching constructed by the non-committal benchmark.

$$\begin{aligned}
&\text{maximize} && \sum_{e \in E} w_e \cdot z_e && && \text{(LP-std-non)}\\
&\text{subject to} && \sum_{\substack{e \in E\\ v \in e}} z_e \leq 1 && \forall v \in V && \text{(C.5)}\\
& && \sum_{\substack{e \in E\\ v \in e}} x_{u,v} \leq \ell_v && \forall v \in V && \text{(C.6)}\\
& && z_e \leq p_e \cdot x_e && \forall e \in E && \text{(C.7)}\\
& && x_e \leq 1 && \forall e \in E && \text{(C.8)}\\
& && x_e, z_e \geq 0 && \forall e \in E && \text{(C.9)}
\end{aligned}$$

We denote $\text{LPOPT}_{std-non}(G)$ as the value of an optimum solution to LP-std-non. It turns out that LP-std and LP-std-non take the same optimum value, no matter the stochastic graph G:

Lemma C.3. *For any stochastic graph $G = (V, E)$,*

$$LPOPT_{std}(G) = LPOPT_{std-non}(G).$$

Proof. Suppose we are presented a solution $(x_e)_{e \in E}$ to LP-std. In this case, if $z_e := p_e \cdot x_e$ for $e \in E$, then $(x_e, z_e)_{e \in E}$ is clearly a feasible solution to LP-std-non. As such,

$$\text{LPOPT}_{std}(G) \leq \text{LPOPT}_{std-non}(G).$$

On the other hand, suppose that $(x_e, z_e)_{e \in E}$ is now an arbitrary solution to LP-std-non. In this case, define $\bar{x}_e := z_e / p_e$ for each $e \in E$. We claim that $(\bar{x}_e)_{e \in E}$ is a feasible solution to LP-std.

To see this, first observe that since $z_e \leq p_e \cdot x_e$, we know that $\bar{x}_e \leq x_e \leq 1$ for all $e \in E$.

Moreover, for each $v \in V$,

$$\sum_{\substack{e \in E:\\ v \in e}} p_e \cdot \bar{x}_e = \sum_{\substack{e \in E:\\ v \in e}} z_e \leq 1,$$

and

$$\sum_{\substack{e \in E:\\ v \in e}} \bar{x}_e \leq \sum_{\substack{e \in E:\\ v \in e}} x_e \leq \ell_v.$$

Thus, $(\bar{x}_e)_{e \in E}$ is a feasible solution to LP-std.

Finally, observe that

$$\sum_{e \in E} w_e \cdot z_e = \sum_{e \in E} p_e \cdot w_e \cdot \bar{x}_e,$$

so $\text{LPOPT}_{std-non}(G) \leq \text{LPOPT}_{std}(G)$.

□

Proof of Theorem C.1. Let us suppose that $\mathcal{M}$ is the matching returned when the non-committal benchmark executes on $G = (V, E)$. If we fix $e \in E$, then we can define x_e as the probability $\mathcal{A}$ probes the edge e, and z_e as the probability that it includes e in $\mathcal{M}$. Observe then that

$$\text{OPT}_{non}(G) = \mathbb{E}[\text{val}(\mathcal{M})] = \sum_{e \in E} w_e \cdot z_e.$$

Now, if we can show that $(x_e, z_e)_{e \in E}$ is a feasible solution to LP-std-non, then this will imply that

$$\text{OPT}_{non}(G) \leq \text{LPOPT}_{std-non}(G) = \text{LPOPT}_{std}(G),$$

where the final line follows via an application of Lemma C.3. Thus, in order to complete the proof it suffices to show that $(x_e, z_e)_{e \in E}$ is a feasible solution to LP-std-non.

Suppose now that we fix vertex $v \in V$. Observe that since it is matched to at most one edge of G, we have that

$$\sum_{\substack{e \in E \\ v \in e}} z_e \leq 1.$$

Similarly,

$$\sum_{\substack{e \in E \\ v \in e}} x_e \leq \ell_v,$$

as at most ℓ_v edges including v are probed by the non-committal benchmark.

If we now fix an edge $e \in E$, then observe that in order for e to be included in $\mathcal{M}$, e must be probed *and* e must be active. On the other hand, these two events occur independently of each other. As such,

$$z_e \leq p_e \cdot z_e.$$

This shows that all the constraints of LP-std-non hold for $(x_e, z_e)_{e \in E}$, and so the proof is complete. □

We conclude the section by observing that Theorem C.2 can be seen to hold by applying Theorem C.1 to the instantiated graph $\hat{G} \sim (G, \boldsymbol{r}, n)$, and using a standard conditioning argument (see the proof of Lemma 4.1 from Section 4 for details).

C.2 LP-new and the Non-committal Benchmark

We begin by observing that since LP-new and LP-std are the same LP when $G = (U, V, E)$ is bipartite and has unit patience values on V, Theorem C.1 implies that the competitive ratios in Theorems 3.1, 3.5 and 5.2 all hold against the *non-committal* benchmark, for the special case of unit patience. Moreover, Theorem C.2 implies that the competitive ratio of Theorem 4.3 holds against the non-committal benchmark when the type graph has unit patience.

In the case when G has full patience, observe that LP-new takes the same value as the LP in [21] considered by Gamlath et al. (see LP-rel in Appendix A for details). On the other hand, Gamlath et al. argue that an optimum solution to their LP upper bounds the expected weight of an optimum matching of G. As G has full patience, this corresponds to the performance of the non-committal benchmark on G, and so LP-new is a relaxation of the non-committal benchmark for this special case. The competitive ratios of Theorems 3.1, 3.5 and 5.2 thus all hold against the non-committal benchmark for full patience. When G is a type graph with full patience, the same techniques used in Lemma 4.1 from Section 4 can be used to show that LP-new-iid upper bounds the non-committal benchmark. This implies that Theorem 4.3 also holds against the non-committal benchmark in this setting.

While our competitive ratios carry over in these specific scenarios, consider now the setting when G has a single online node v. In this case, it is clear that LP-new exactly encodes the committal benchmark; that is, $\text{LPOPT}_{new}(G) = \text{OPT}(G)$. Moreover, $\text{OPT}(G)$ can be attained via a non-adaptive strategy. On the other hand, the edges probabilities and edge weights of G can be chosen in such a way that the non-committal benchmarks gains strictly more power than the committal benchmark; that is, $\text{OPT}(G) < \text{OPT}_{non}(G)$. We remark that these problems are a special case of ProblemMax, a stochastic probing problem which is studied in [5, 20, 36].

Let us suppose that v has patience $\ell_v = 2$, and that there are 3 offline nodes $U = \{u_1, u_2, u_3\}$. For each $i \in \{1, 2, 3\}$, we denote the weight of (u_i, v) by w_i and assume that the edge (u_i, v) is active with probability p_i. We make the following assumptions on these weights and probabilities:

1. $w_1 < w_2 < w_3$.
2. $p_1 > p_2 > p_3$.
3. $w_1 \cdot p_1 \geq w_2 \cdot p_2 > w_3 \cdot p_3$.
4. $p_2 \cdot w_2 - p_3 \cdot w_3 \geq p_1 \cdot w_1 \cdot (p_2 - p_3)$

Clearly, there exists a choice of weights and probabilities which satisfy these constraints. For instance, take $w_1 = 3$, $w_2 = 4$, $w_3 = 98$, $p_1 = 0.8$, $p_2 = 0.6$, and $p_3 = 0.01$.

Based on these assumptions let us now consider the value of $\text{OPT}(G)$. Observe that

$$p_2 \cdot w_2 + (1 - p_2) \cdot p_1 \cdot w_1 \geq p_3 \cdot w_3 + (1 - p_3) \cdot p_1 \cdot w_1 \geq p_3 \cdot w_3 + (1 - p_3) \cdot p_2 \cdot w_2,$$

where the first inequality follows from (4), and the second follows from (3). As a result, it is clear to see that the committal benchmark corresponds to probing u_2 and then u_1 (if necessary); thus, $\text{OPT}(G) = p_2 \cdot w_2 + (1 - p_2) \cdot p_1 \cdot w_1$.

On the other hand, let us consider $\text{OPT}_{non}(G)$, the value of the non-committal benchmark on G. Consider the following *non-committal* probing algorithm:

- Probe (u_2, v), and if $\text{st}(u_2, v) = 1$, probe (u_3, v).
- Else if $\text{st}(u_2, v) = 0$, probe (u_1, v).
- Return the edge of highest weight which is active (if any).

Clearly, this probing algorithm uses adaptivity to decide whether to reveal (u_3, v) or (u_1, v) in its second probe. Specifically, if it discovers that (u_2, v) is active, then it knows that it will return an edge with weight at least w_2. As such, it only makes sense for (u_3, v) to be its next probe, as $w_3 > w_2 > w_1$. On the other hand, if (u_2, v) is discovered to be inactive, it makes sense to prioritize probing the edge (u_1, v) over (u_3, v), as the *expected* reward is higher; namely, $w_1 \cdot p_1 > w_3 \cdot p_3$.

The expected value of the edge returned is

$$p_2 \cdot p_3 \cdot w_3 + p_2 \cdot (1 - p_3) \cdot w_2 + (1 - p_2) \cdot p_1 \cdot w_1.$$

Observe however that

$$\begin{aligned} p_2 \cdot p_3 \cdot w_3 + p_2 \cdot (1 - p_3) \cdot w_2 + (1 - p_2) \cdot p_1 \cdot w_1 &= p_2 \cdot w_2 \cdot ((1 - p_3) + p_3 \cdot w_3/w_2) + (1 - p_2) \cdot p_1 \cdot w_1 \\ &> p_2 \cdot w_2 + (1 - p_2) \cdot p_1 \cdot w_1 \end{aligned}$$

$$= \mathrm{OPT}(G),$$

where the final inequality follows since $w_3 > w_2$. As a result, it is clear that this strategy corresponds to the non-committal benchmark, and so $\mathrm{OPT}_{non}(G) > \mathrm{OPT}(G)$. In fact, for the specific choice when $w_1 = 3$, $w_2 = 4$, $w_3 = 98$, $p_1 = 0.8$, $p_2 = 0.6$, and $p_3 = 0.01$, it holds that

$$\frac{\mathrm{LPOPT}_{new}(G)}{\mathrm{OPT}_{non}(G)} = \frac{\mathrm{OPT}(G)}{\mathrm{OPT}_{non}(G)} = 0.856269.$$

This example illustrates that since our competitive guarantees were proven against $\mathrm{LPOPT}_{new}(G)$, they do not immediately extend to the non-committal benchmark. Moreover, this ratio improves upon the negative result of [13], in which the authors present an example where the ratio between $\mathrm{OPT}(G)$ and $\mathrm{OPT}_{non}(G)$ is at most 0.898.

D LP Relations

In this section, we show how a number of the LPs present in the literature are related to each other[18]. In particular, we consider an LP introduced in by Brubach et al. [9], which assumes a number of extra constraints in addition to those of LP-std. We review the motivation behind this LP, as well as how it is derived.

For each subset $R \subseteq U$ and $v \in V$, consider the induced stochastic subgraph, denoted $G[\{v\} \cup R]$, formed by restricting the vertices of G to $\{v\} \cup R$, and the edges of G to those between v and R. We hereby denote $\mathrm{OPT}(v, R)$ as the value of the committal benchmark on the induced stochastic graph $G[\{v\} \cup R]$.

We can now formulate the LP of [9], whose constraints ensure that for each $v \in V$, the expected stochastic reward of v, suggested by an LP solution, is actually attainable by the committal benchmark.

$$\begin{aligned}
&\text{maximize} && \sum_{u \in U} \sum_{v \in V} w_{u,v}\, p_{u,v}\, x_{u,v} && && \text{(LP-DP)}\\
&\text{subject to} && \sum_{v \in V} p_{u,v}\, x_{u,v} \leq 1 && \forall u \in U && \text{(D.1)}\\
& && \sum_{u \in U} x_{u,v} \leq \ell_v && \forall v \in V && \text{(D.2)}\\
& && \sum_{u \in U} p_{u,v}\, x_{u,v} \leq 1 && \forall v \in V && \text{(D.3)}\\
& && \sum_{u \in R} w_{u,v}\, p_{u,v}\, x_{u,v} \leq \mathrm{OPT}(v, R) && \forall v \in V,\ R \subseteq U && \text{(D.4)}\\
& && 0 \leq x_{u,v} \leq 1 && \forall u \in U, v \in V && \text{(D.5)}
\end{aligned}$$

Observe the following relations between the LPs considered in the paper.

Theorem D.1. *For any stochastic graph G, we have that*

$$LPOPT_{new}(G) \leq LPOPT_{DP}(G) \leq LPOPT_{std}(G), \tag{D.6}$$

and the LPs are all equivalent, provided G has unit patience.

[18] We do not consider the full patience LP of Gamlath et al. in this section, as we discuss a generalization of their LP in Appendix C.

The second inequality is immediate since LP-DP is a tightening of LP-std. To prove the first inequality, we will first proceed to state and prove Lemma D.2.

Assume that for each $u \in U$ and $v \in V$, we are presented a fractional value, $0 \leq x_{u,v} \leq 1$. Moreover, let us assume that the values $(x_{u,v})_{u \in U, v \in V}$ satisfy the following properties:

1. For each $u \in U$,
$$\sum_{v \in V} p_{u,v} \, x_{u,v} \leq 1. \tag{D.7}$$
2. For each $v \in V$, there exists a probing algorithm $\mathcal{A}_v$ for the instance $G[\{v\} \cup U]$ which respects commitment and for which
$$\mathbb{P}[\mathcal{A}_v \text{ probes } (u,v)] = x_{u,v}, \tag{D.8}$$
for each $u \in U$ [19].

In this case, we get the following lemma:

Lemma D.2. *If the values $(x_{u,v})_{u \in U, v \in V}$ satisfy properties* (D.7) *and* (D.8)*, then $(x_{u,v})_{u \in U, v \in V}$ is a feasible solution to LP-DP.*

Proof. Let us fix $v \in V$. We first observe that

$$\sum_{u \in U} \mathbb{P}[\mathcal{A}_v \text{ probes } (u,v)]$$

corresponds to the expected number of probes that $\mathcal{A}_v$ makes when executing on $G[\{v\} \cup U]$. Thus, since $\mathcal{A}_v$ makes at most ℓ_v probes, we know that

$$\sum_{u \in U} x_{u,v} = \sum_{u \in U} \mathbb{P}[\mathcal{A}_v \text{ probes } (u,v)] \leq \ell_v.$$

Let us now denote $\mathcal{M}$ as the matching returned once $\mathcal{A}_v$ finishes executing on $G[\{v\} \cup U]$. This is either a single edge including v, or the empty-set. As such, we denote $\mathcal{M}(v)$ to indicate which vertex v is matched to (where $\mathcal{M}(v) := \emptyset$ if v remains unmatched).

Observe then that for each $u \in U$, we have that

$$\begin{aligned} \mathbb{P}[\mathcal{M}(v) = u] &= \mathbb{P}[\mathcal{A}_v \text{ probes } (u,v) \text{ and } \mathrm{st}(u,v) = 1] \quad &\text{(D.9)} \\ &= p_{u,v} \, x_{u,v}, &\text{(D.10)} \end{aligned}$$

as $\mathcal{A}_v$ respects commitment by assumption.

As a result, since v is matched to at most one vertex of U,

$$\sum_{u \in U} p_{u,v} \, x_{u,v} \leq 1.$$

It remains to verify that the additional LP constraints present in LP-DP hold for v. Let us define $\mathrm{val}(\mathcal{M})$ as the weight of the edge matched to v (which is 0 if v remains unmatched by $\mathcal{A}_v$). Observe that

$$\mathbb{E}[\mathrm{val}(\mathcal{M})] = \sum_{u \in U} w_{u,v} \, p_{u,v} \, x_{u,v},$$

[19] In the terminology of Section 2, we say that the values $(x_{u,v})_{u \in U, v \in V}$ can be implemented **losslessly**.

after applying (D.9) and linearity of expectation. Thus, since $\mathcal{A}_v$ is a valid probing algorithm which executes on $G[\{v\} \cup U]$, this value can be no larger than what is attained by the committal benchmark on $G[\{v\} \cup U]$. As such,

$$\sum_{u \in U} w_{u,v}\, p_{u,v}\, x_{u,v} = \mathbb{E}[\text{val}(\mathcal{M})] \leq \text{OPT}(v, U),$$

where $\text{OPT}(v, U)$ corresponds to the value of the committal benchmark on $G[\{v\} \cup U]$. More generally, if we now fix $R \subseteq U$, then observe that

$$\sum_{u \in R} w_{u,v}\, p_{u,v}\, x_{u,v} = \mathbb{E}[\text{val}(\mathcal{M})\, \mathbf{1}_{[\mathcal{M}(v) \in R]}],$$

where $\mathcal{M}(v) \in R$ corresponds to the event in which the vertex matched to v lies in R.

Of course, we can also modify the probing algorithm $\mathcal{A}_v$ in such a way that it returns $\emptyset$ instead an edge within $\overline{R} \times \{v\}$. This alternative probing algorithm (which depends on R) will then return an edge of expected value

$$\mathbb{E}[\text{val}(\mathcal{M})\, \mathbf{1}_{[\mathcal{M}(v) \in R]}].$$

As such, for each $R \subseteq U$,

$$\sum_{u \in R} w_{u,v}\, p_{u,v}\, x_{u,v} = \mathbb{E}[\text{val}(\mathcal{M})\, \mathbf{1}_{[\mathcal{M}(v) \in R]}] \leq \text{OPT}(v, R),$$

where $\text{OPT}(v, R)$ corresponds to the committal benchmark on $G[\{v\} \cup R]$.

Now, the vertex v was arbitrary, so we know that all the constraints on the vertices of LP-DP hold. By assumption, we also know that for each $u \in U$,

$$\sum_{v \in V} p_{u,v}\, x_{u,v} \leq 1.$$

Thus, $(x_{u,v})_{u \in U, v \in V}$ is a feasible solution to LP-DP, thereby completing the proof.

□

With this lemma, we now prove Theorem D.1.

Proof of Theorem D.1. Suppose that we are presented an optimum solution to LP-new, denoted $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$. Recall that for each $u \in U, v \in V$ and we defined the edge variable $x_{u,v}$, where

$$\widetilde{x}_{u,v} = \sum_{i=1}^{\ell_v} \sum_{\substack{\boldsymbol{u}^* \in U^{(\leq \ell_v)}: \\ u_i^* = u}} \frac{g_v^i(\boldsymbol{u}^*)\, x_v(\boldsymbol{u}^*)}{p_{u,v}}.$$

We first observe that the values $(\widetilde{x}_{u,v})_{u \in U, v \in V}$ satisfy property (D.7) by assumption (see (2.8) of LP-new). Moreover, for each fixed $v \in V$, if we consider the values $(\widetilde{x}_{u,v})_{u \in U}$, then the VertexProbe algorithm applied to the input $(G, (x_v(\boldsymbol{u}))_{\boldsymbol{u} \in U^{(\leq \ell_v)}}, v)$, satisfies property (D.8), by Lemma 2.3.

We may therefore conclude that $(\widetilde{x}_{u,v})_{u \in U, v \in V}$ is a feasible solution to LP-DP. On the other hand, $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$ is an optimum solution to LP-new, so

$$\text{LPOPT}_{new}(G) = \sum_{u \in U, v \in V} w_{u,v}\, p_{u,v}\, \widetilde{x}_{u,v} \leq \text{LPOPT}_{DP}(G),$$

thus proving Theorem D.1.

□

We conclude the section by observing the following relation between the LPs in the known i.i.d. stochastic matching setting, namely LP-new-iid and LP-std-iid:

Proposition D.3. *If $(G, \boldsymbol{r}, n)$ is a known i.i.d. input, then*

$$LPOPT_{new-iid}(G, \boldsymbol{r}, n) \leq LPOPT_{std-iid}(G, \boldsymbol{r}, n).$$

In fact, LP-new-iid and LP-std-iid are identical when G has unit patience.

This follows via a standard conditioning argument involving the instantiated graph $\hat{G} \sim (G, \boldsymbol{r}, n)$, combined with an application of Theorem D.1, so we omit the argument.

E Non-adaptive Probing Algorithms and Adaptivity Gaps

Suppose that $G = (U, V, E)$ is an arbitrary stochastic graph, and we are presented an online probing algorithm $\mathcal{A}$ which is non-adaptive and respects commitment (as defined in Section 2). We may assume that $\mathcal{A}$ operates in the ROM setting, that is the ordering π on V is chosen uniformly at random, though the definitions we now describe follow identically when π is chosen by an adversary, as well as in the known i.i.d. setting.

We hereby denote $\mathcal{A}(G)$ as the matching returned by executing $\mathcal{A}$ on G, and $\text{val}(\mathcal{A}(G))$ as the (random) value of this matching.

With this notation, we define the *adaptivity gap* of a stochastic graph G in the ROM setting as the ratio,

$$\frac{\sup_{\mathcal{B}} \mathbb{E}[\text{val}(\mathcal{B}(G))]}{\text{OPT}(G)},$$

where the supremum is over all non-adaptive online probing algorithms.

While all of the algorithms we consider throughout the paper are implemented non-adaptivity, of particular interest to us are Algorithms 3 and 5 in which the stochastic graph G is presented ahead of time. Observe that Theorems 3.1 and 3.8, imply the following bounds on the relevant adaptivity gaps:

Corollary E.1. *The known stochastic matching problem with offline vertex weights, arbitrary patience and adversarial arrivals has an adaptivity gap no worse than $1 - 1/e$.*

Corollary E.2. *The known stochastic matching problem with arbitrary patience, edge weights and ROM arrivals has an adaptivity gap which is no worse than $1 - 1/e$.*

F Deferred Proofs

Proof of Theorem 3.5. In this setting, the order of online vertices π is generated uniformly at random. As such, we denote the vertices of V as $v_1, \ldots, v_n$, where v_t corresponds to the vertex in position $1 \leq t \leq n$ of π (and $n := |V|$).

For each $u \in U$ and $v \in V$, we once again make use of the edge variables $(\widetilde{x}_{u,v})_{u \in U, v \in V}$ associated to the solution $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\ell_v)}}$.

Let us now fix a particular vertex $v \in V$, and a vertex $u \in U$. We say that u is *free* for v, provided u is unmatched when v is processed by Algorithm 3.

Observe then that if $C(u, v)$ corresponds to the event in which v commits to u during one of its ℓ_v probes, then

$$\mathbb{P}[\mathcal{M}(v) = u] = \mathbb{P}[C(u, v) \text{ and } u \text{ is free for } v]$$

$$
\begin{aligned}
&= \mathbb{P}[C(u,v)] \cdot \mathbb{P}[u \text{ is free for } v] \\
&= p_{u,v}\, \widetilde{x}_{u,v}\, \mathbb{P}[u \text{ is free for } v],
\end{aligned}
$$

where the final line follows from Lemma 2.3.

We know however that,

$$
\begin{aligned}
\mathbb{P}[u \text{ is free for } v] &= \sum_{t=1}^{n} \mathbb{P}[u \text{ is free for } v \,|\, v_t = v] \cdot \mathbb{P}[v_t = v] && \text{(F.1)} \\
&= \sum_{t=1}^{n} \frac{\mathbb{P}[u \text{ is free for } v \,|\, v_t = v]}{n}, && \text{(F.2)}
\end{aligned}
$$

where the last equality follows since π is generated uniformly at random.

As such, we may lower bound $\mathbb{P}[u \text{ is free for } v \,|\, v_t = v]$ for each $t = 1, \ldots, n$ in order to derive a lower bound on the competitive ratio of the algorithm.

Let us now fix $1 \leq t \leq n$ and condition on the event in which $v_t = v$. Observe then that

$$
\mathbb{P}[u \text{ is not free for } v_t \,|\, v_t = v] = \mathbb{P}[\cup_{k=1}^{t-1} \mathcal{M}(v_k) = u \,|\, v_t = v] \leq \sum_{k=1}^{t-1} \mathbb{P}[\mathcal{M}(v_k) = u \,|\, v_t = v],
$$

as u is not free for v_t, if and only if one of $v_1, \ldots, v_{t-1}$ matches to u.

On the other hand, using Lemma 2.3, we know that for each $k = 1, \ldots, t-1$

$$
\begin{aligned}
\mathbb{P}[\mathcal{M}(v_k) = u \,|\, v_t = v] &= \sum_{\substack{s \in V: \\ s \neq v}} \mathbb{P}[\mathcal{M}(s) = u \,|\, \{v_t = v\} \cap \{v_k = s\}] \cdot \mathbb{P}[v_k = s \,|\, v_t = v] \\
&\leq \sum_{\substack{s \in V: \\ s \neq v}} \mathbb{P}[C(s,u) \,|\, \{v_t = v\} \cap \{v_k = s\}] \cdot \mathbb{P}[v_k = s \,|\, v_t = v] \\
&= \sum_{\substack{s \in V: \\ s \neq v}} \frac{p_{u,s}\, \widetilde{x}_{u,s}}{n-1},
\end{aligned}
$$

as once we condition on $\{v_t = v\}$, v_k is uniformly distributed amongst $V \setminus \{v\}$.

As a result,

$$
\mathbb{P}[u \text{ is not free for } v_t \,|\, v_t = v] \leq (t-1) \sum_{\substack{s \in V: \\ s \neq v}} \frac{p_{u,s}\, \widetilde{x}_{u,s}}{n-1} \leq \frac{t-1}{n-1},
$$

by the constraints of LP-new.

Thus, combined with (F.1),

$$
\begin{aligned}
\mathbb{P}[u \text{ is free for } v] &\geq \sum_{t=1}^{n} \frac{1}{n} \left(1 - \frac{t-1}{n-1}\right) \\
&= 1 - \frac{\sum_{t=1}^{n}(t-1)}{n\,(n-1)} \\
&= 1/2.
\end{aligned}
$$

To conclude, for each edge $(u, v) \in E$, we have that

$$\mathbb{P}[\mathcal{M}(v) = u] = p_{u,v}\, \widetilde{x}_{u,v}\, \mathbb{P}[u \text{ is free for } v] \geq \frac{p_{u,v}\, \widetilde{x}_{u,v}}{2}.$$

On the other hand, if we denote $\text{val}(\mathcal{M})$ as the value of the matching $\mathcal{M}$, then $\text{val}(\mathcal{M}) = \sum_{u \in U, v \in V} w_{u,v}\, \mathbf{1}_{[\mathcal{M}(v)=u]}$. Thus,

$$\mathbb{E}[\text{val}(\mathcal{M})] = \sum_{u \in U, v \in V} w_{u,v}\, \mathbb{P}[\mathcal{M}(v) = u] \geq \sum_{u \in U, v \in V} \frac{w_{u,v}\, \widetilde{x}_{u,v}\, p_{u,v}}{2}.$$

As $(x_v(\boldsymbol{u}))_{v \in V, \boldsymbol{u} \in U^{(\leq \ell_v)}}$ is an optimum solution to LP-new, this completes the proof. □

Proof of Lemma 4.1. Suppose that $(G, \boldsymbol{r}, n)$ is a known i.i.d. instance, where $G = (U, V, E)$ is a type graph with maximum patience $\ell := \max_{v \in V} \ell_v$. We can then define the following collection of random variables, denoted $(X_t(\boldsymbol{u}))_{t \in [n], \boldsymbol{u} \in U^{(\leq \ell)}}$, based on the following randomized procedure:

- Draw the instantiated graph $\hat{G} \sim (G, \boldsymbol{r}, n)$, whose vertex arrivals we denote by $v_1, \ldots, v_n$.
- Compute an optimum solution of LP-new for $\hat{G}$, which we denote by $(x_{v_t}(\boldsymbol{u}))_{t \in [n], \boldsymbol{u} \in U^{(\leq v_t)}}$.
- For each $t = 1, \ldots, n$ and $\boldsymbol{u} \in U^{(\leq \ell)}$, set $X_t(\boldsymbol{u}) = x_{v_t}(\boldsymbol{u})$ if $\boldsymbol{u} \in U^{(\leq \ell_{v_t})}$, otherwise set $X_t(\boldsymbol{u}) = 0$.

Observe then that by definition, $(X_t(\boldsymbol{u}))_{t \in [n], \boldsymbol{u} \in U^{(\leq \ell_{v_t})}}$ is a feasible solution to LP-new for $\hat{G}$. As such, for each $t = 1, \ldots, n$

$$\sum_{\boldsymbol{u} \in U^{(\leq \ell)}} X_t(\boldsymbol{u}) \leq 1, \tag{F.3}$$

and for each $u \in U$,

$$\sum_{t \in [n], i \in [\ell]} \sum_{\substack{\boldsymbol{u}^* \in U^{(\leq \ell)}: \\ u_i^* = u}} g_v^i(\boldsymbol{u}^*) \cdot X_t(\boldsymbol{u}^*) \leq 1. \tag{F.4}$$

Moreover, $(X_t(\boldsymbol{u}))_{t \in [n], \boldsymbol{u} \in U^{(\leq \ell_{v_t})}}$ is a optimum solution to LP-new for $\hat{G}$, and so Theorem 2.2 implies that

$$\text{OPT}(\hat{G}) \leq \text{LPOPT}_{new}(\hat{G}) = \sum_{t=1}^{n} \sum_{\boldsymbol{u} \in U^{(\leq \ell)}} \left(\sum_{i=1}^{|\boldsymbol{u}|} w_{u_i, v} \cdot g_v^i(\boldsymbol{u}) \right) \cdot X_t(\boldsymbol{u}). \tag{F.5}$$

In order to make use of these inequalities in the context of the type graph G, let us first fix a type node $v \in V$ and a tuple $\boldsymbol{u} \in U^{(\leq \ell)}$. We can then define

$$y_v(\boldsymbol{u}) := \sum_{t=1}^{n} \mathbb{E}[X_t(\boldsymbol{u}) \cdot \mathbf{1}_{[v_t = v]}], \tag{F.6}$$

where the randomness is over the generation of $\hat{G}$. Observe that by definition of the $(X_t(\boldsymbol{u}))_{t \in [n], \boldsymbol{u} \in U^{(\leq \ell)}}$ values,

$$y_v(\boldsymbol{u}) = 0,$$

provided $|\boldsymbol{u}| > \ell_v$.

We claim that $(y_v(\boldsymbol{u}))_{v\in V, \boldsymbol{u}\in U^{(\le \ell_v)}}$ is a feasible solution to LP-new-iid. To see this, first observe that if we multiply (F.3) by the indicator random variable $\mathbf{1}_{[v_t=v]}$ while summing over $t \in [n]$, then we get that

$$\sum_{\boldsymbol{u}^*\in U^{(\le \ell)}} \sum_{t=1}^{n} X_t(\boldsymbol{u}) \cdot \mathbf{1}_{[v_t=v]} \le \sum_{t=1}^{n} \mathbf{1}_{[v_t=v]}.$$

As a result, if we take expectations over this inequality,

$$\begin{aligned}\sum_{\boldsymbol{u}\in U^{(\le \ell)}} y_v(\boldsymbol{u}) &= \sum_{\boldsymbol{u}\in U^{(\le \ell)}} \mathbb{E}\left[\sum_{t=1}^{n} X_t(\boldsymbol{u}) \cdot \mathbf{1}_{[v_t=v]}\right] \\ &\le \sum_{t=1}^{n} \mathbb{P}[v_t = v] \\ &= r_v,\end{aligned}$$

for each $v \in V$.

Let us now fix $u \in U$. Observe that by rearranging the left-hand side of (F.4),

$$\sum_{t\in[n]} \sum_{i\in[\ell]} \sum_{\substack{\boldsymbol{u}^*\in U^{(\le \ell)}: \\ u^*_i = u}} g^i_v(\boldsymbol{u}^*) \cdot X_t(\boldsymbol{u}^*) = \sum_{i\in[\ell]} \sum_{\substack{\boldsymbol{u}^*\in U^{(\le \ell)}: \\ u^*_i = u}} g^i_v(\boldsymbol{u}) \cdot \sum_{v\in V} \sum_{t\in[n]} X_t(\boldsymbol{u}^*) \cdot \mathbf{1}_{[v_t=v]}. \tag{F.7}$$

Thus, after taking expectation over (F.4),

$$\sum_{v\in V} \sum_{i\in[\ell]} \sum_{\substack{\boldsymbol{u}^*\in U^{(\le \ell_v)}: \\ u^*_i = u}} y_v(\boldsymbol{u}^*) \le 1,$$

for each $u \in U$.

Since $(y_v(\boldsymbol{u}))_{v\in V, \boldsymbol{u}\in U^{(\le \ell_v)}}$ satisfies these inequalities, and the variables are clearly all non-negative, it follows that $(y_v(\boldsymbol{u}))_{v\in V, \boldsymbol{u}\in U^{(\le \ell_v)}}$ is a feasible solution to LP-new-iid.

In order to complete the proof, let us rearrange the right-hand side of (F.5) as in (F.7) and take expectations. We then get that

$$\mathbb{E}[\mathrm{OPT}(\hat{G})] \le \sum_{v\in V} \sum_{\boldsymbol{u}\in U^{(\le \ell)}} \left(\sum_{i=1}^{|\boldsymbol{u}|} w_{u_i, v} \cdot g^i_v(\boldsymbol{u}) \right) \cdot y_v(\boldsymbol{u}).$$

Now, $\mathrm{OPT}(G, \boldsymbol{r}, n) = \mathbb{E}[\mathrm{OPT}(\hat{G})]$ by definition, so since $(y_v(\boldsymbol{u}))_{v\in V, \boldsymbol{u}\in U^{(\le \ell_v)}}$ is feasible, it holds that

$$\mathrm{OPT}(G, \boldsymbol{r}, n) \le \mathrm{LPOPT}_{new-iid}(G, \boldsymbol{r}, n),$$

thus completing the proof.

□

G Extended Related Works

Our results pertain to the online stochastic matching problem which (loosely speaking) is online bipartite matching where edges are associated with their probabilities of existence. There is a substantial body of research pertaining to the "classical" (i.e. non stochastic) online bipartite model in the fully adversarial online model, the random order model, and the i.i.d. input model. The ever growing interest in various online bipartite matching problems is a reflection of the importance of online advertising but there are many other natural applications. The literature concerning competitive analysis[20] of online bipartite matching is too extensive to do justice to many important papers. We refer the reader to the excellent 2013 survey by Mehta [33] with emphasis on online variants relating to ad-allocation. Given the continuing interest in ad-allocation, the survey is not current but does describe the basic results.

The seminal result for unweighted online bipartite matching is due to Karp, Vazirani, and Vazirani [28]. They gave the randomized Ranking algorithm that achieves competitive ratio $1-1/e$ in the adversarial online setting which they show is the best possible ratio for any randomized algorithm. There have been many proofs of this seminal result, such as the primal-dual approach due to Devanur et al. [16]. Any greedy algorithm (i.e., one that always makes a match when possible) has a 0.5 ratio, and this is the best possible a deterministic algorithm can attain. The Ranking algorithm can also be viewed as a deterministic algorithm in the ROM input model. In the ROM model, Madhian and Yan [31] show that the randomized Ranking algorithm achieves competitive ratio .696. For the case of weighted offline vertices and adversarial input sequences, Aggarwal et al. [4] were able to achieve a randomized $1-1/e$ competitive ratio by their Perturbed Ranking algorithm. Huang et al. [25] show that the Perturbed Ranking algorithm obtains a .6534 competitive ratio in the ROM input model.

Feldman et al. [19] introduced online bipartite matching in the i.i.d. model in which each online vertex is independently and identically generated from some known distribution. In this model, they were able to beat the $1-1/e$ inapproximation for bipartite matching that applies to the fully adversarial online model. The i.i.d. online bipartite model has been studied for the unweighted and edge weighted models. The most recent competitive ratios for integral arrival rates are due to Brubach et al. [10] in which they derive a .7299 ratio for the (offline) vertex weighted case and a .705 ratio for edge weighted graphs. Karande et al. [27] show that any competitive ratio for the ROM model applies to the unknown (and therefore known) i.i.d. models. It follows that any inapproximation for the known i.i.d. model applies to the ROM model. Kesselheim et al. [29] extend the classical secretary result and established the optimal $1/e$ ROM ratio for bipartite matching with edge weights.

An early example of stochastic probing without commitment is the Pandora's box problem attributed to Weitzman [39]. In Weitzman's Pandora's box problem, a set of boxes is given, where each box contains a stochastic value from a known distribution and a cost for opening (i.e., probing) the box. The algorithm has the option at any time of accepting the value of any opened box and pays the total cost of all opened boxes. This is an offline probing problem in that boxes can be opened in any order. An online version of the Pandora's box problem has recently been studied in Esfandiari et al. [18]. Stochastic probing with commitment has been studied for various packing problems, most notably for the knapsack problem, as studied in Dean et al. [14, 15]. In the

[20] Initially, competitive analysis refered to the relative performance (i.e., the competitive ratio) of an online algorithm as compared to an optimal solution (in the worst case over all input sequences determined adversarially). We extend the meaning of the competitive ratio to also refer to input sequences generated in the ROM model as well as sequences generated i.i.d. from a known or unknown distribution; that is, whenever the algorithm has no control over the order of input arrivals.

stochastic knapsack setting, the stochastic inputs are items whose values are known but whose sizes are stochastic and not known until the algorithm probes the item. As soon as the knapsack capacity is exceeded by a probed item, the algorithm terminates. Dean et al. also introduced the offline issue of measuring the benefit of adaptively choosing probes versus having a fixed order of probes.

Turning back to matching problems, Chen et al. [12] introduced the stochastic matching problem assuming a known stochastic graph and algorithms that can probe any edge in any order. They obtained a 4-approximation[21] greedy algorithm in the unweighted case for arbitrary patience values. They conjectured that their greedy algorithm was a 2-approximation. Subsequently, Adamczyk [1] confirmed that the greedy algorithm is a 2-approximation for the unweighted problem and that this approximation is tight. Bansal et al. [6] established a 4-approximation for the edge weighted case with arbitrary patience and a 3-approximation for the special case of bipartite graphs. Adamczyk et al. [3] improved the Bansal et al. bounds providing an approximation algorithm with a ratio of 2.845 for bipartite graphs and an algorithm with a ratio of 3.709 for general graphs. Baveja et al. [7] recently improved the analysis of the original algorithm of Bansal et al., yielding an approximation ratio of 3.224 for general graphs.

Of particular importance to our paper is the known stochastic matching framework with ROM arrivals, as defined precisely in Section 2. Gamlath et al. [21] presented a probing algorithm which is a $1-\frac{1}{e}$-approximation for the bipartite case in the *full patience setting*; that is, when there are no patience restrictions for nodes on either side of the bipartition. The full patience setting is closely related to the bipartite matching algorithm studied by Ehsani et al. [17], which they prove is a $1-\frac{1}{e}$-approximation as a corollary of their work in the more general *combinatorial auctions prophet secretary problem.* While not explicitly stated in [17], their bipartite matching algorithm can be interpreted as an adaptive probing algorithm in the known stochastic matching framework with ROM arrivals, attaining the same $1-\frac{1}{e}$ non-adaptive approximation ratio as Gamlath et al.. Very recently, Tang et al. [37] provided an alternative algorithm also attaining the same approximation ratio of $1-\frac{1}{e}$ in the more general *oblivious bipartite matching* setting, however their algorithm does not execute in an online fashion, and so is incomparable. See also Tang et al. [38] for an online greedy algorithm achieving a .501 ratio for a known stochastic graph with edge weights.

Mehta and Panigrahi [34] adapted the stochastic matching problem to the online setting problem with unit patience where the stochastic graph is not known to the algorithm. They specifically considered the unweighted case for unit patience (for the online nodes) and uniform edge probabilities (i.e,, for every edge e, $p_e = p$ for some fixed probability p). They showed that every greedy algorithm has competitive ratio $\frac{1}{2}$. In the same online setting, they provided a greedy algorithm that achieves competitive ratio $\frac{1}{2}(1+(1-p)^{2/p})$ which limits to $\frac{1}{2}(1+e^{-2}) \approx .567$ as $p \to 0$. They also show that against a "standard linear programming (LP)" benchmark, that the best possible ratio is $.621 < 1-\frac{1}{e}$. However, this does not preclude a $1-\frac{1}{e}$ competitive ratio for a stricter LP bound on an optimal stochastic probing algorithm. Preceding the Mehta and Panigrahi work is a result in Bansal et al. [6] where they consider a known stochastic (type) graph with a distribution on the online nodes. This can be called the stochastic matching problem with known i.i.d. inputs. Bansal et al. achieve a 7.92 competitive ratio (or approximately, .13 as a fraction) in this stochastic i.i.d. model. This was improved to .24 by Adamczyk [3] and most recently, by Brubach et al. [11] where they obtain a .46 competitive ratio and a $1-\frac{1}{e}$ inapproximation against a standard LP.

Returning to the unknown stochastic graph setting, there are recent independent papers by

[21] Unfortunately, approximation and competitive bounds for maximization problems are sometimes represented both as ratios > 1 and as fractions < 1. We shall report these ratios as stated in the relevant papers. Our results will be stated as fractions.

Goyal and Udwani [24] and Brubach et al. [9]. Goyal and Udwani consider the vertex weighted unit patience problem and establish a (best possible) $1-\frac{1}{e}$ competitive ratio against an LP that acts as an upper bound on the committal benchmark under the assumption that the edge probabilities are decomposable (i.e., $p_{u,v} = p_u \cdot p_v$) and a .596 competitive ratio for vanishingly small edge probabilities. Our paper is motivated by and most closely follows the Brubach et al. [9] paper. Brubach et al. use and motivate the "ideal stochastic benchmark" (for arbitrary patience) and an LP relaxation for that ideal benchmark. They establish a best possible deterministic $\frac{1}{2}$ competitive ratio against their LP for the vertex weighted online stochastic matching problem. In a recent paper, Huang and Zhang [26] provide a randomized algorithm for unit patience and offline vertex weights in the online stochastic matching framework. In the limit as edge probabilities decrease, their algorithm achieves a .572 competitive ratio.