Prophet Matching in the Probe-Commit Model

Allan Borodin
Department of Computer Science, University of Toronto, Canada

Calum MacRury
Department of Computer Science, University of Toronto, Canada

Akash Rakheja
Department of Computer Science, University of Toronto, Canada

Abstract

We consider the online bipartite stochastic matching problem with known i.d. (independently distributed) online vertex arrivals. In this problem, when an online vertex arrives, its weighted edges must be probed (queried) to determine if they exist, based on known edge probabilities. Our algorithms operate in the probe-commit model, in that if a probed edge exists, it must be used in the matching. Additionally, each online node has a downward-closed probing constraint on its adjacent edges which indicates which sequences of edge probes are allowable. Our setting generalizes the commonly studied patience (or time-out) constraint which limits the number of probes that can be made to an online node's adjacent edges. Most notably, this includes non-uniform edge probing costs (specified by knapsack/budget constraint). We extend a recently introduced configuration LP to the known i.d. setting, and also provide the first proof that it is a relaxation of an optimal offline probing algorithm (the offline adaptive benchmark). Using this LP, we establish the following competitive ratio results against the offline adaptive benchmark:

1. A tight $\frac{1}{2}$ ratio when the arrival ordering π is chosen adversarially.
2. A $1 - 1/e$ ratio when the arrival ordering π is chosen u.a.r. (uniformly at random).

If π is generated adversarially, we generalize the prophet inequality matching problem. If π is u.a.r., we generalize the prophet secretary matching problem. Both results improve upon the previous best competitive ratio of 0.46 in the more restricted known i.i.d. (independent and identically distributed) arrival model against the standard offline adaptive benchmark due to Brubach et al. We are the first to study the prophet secretary matching problem in the context of probing, and our $1 - 1/e$ ratio matches the best known result without probing due to Ehsani et al. This result also applies to the unconstrained bipartite matching probe-commit problem, where we match the best known result due to Gamlath et al.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Stochastic probing, Online algorithms, Bipartite matching, Optimization under uncertainty

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2022.46

Category APPROX

Related Version *Full Version*: `https://arxiv.org/abs/2102.04325` [9]

Acknowledgements We would like to thank Brendan Lucier and David Wajc for their constructive comments on an early version of this paper.

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, online advertising, and kidney exchange applications). There is by now an extensive literature for stochastic probing problems.

Although we are only considering "one-sided online bipartite matching", stochastic matching was first considered in the context of a general graph by Chen et al. [18]. In this problem, the algorithm is presented an adversarially generated *stochastic graph* $G = (V, E)$ as input, which has a probability p_e associated with each edge e and a *patience* (or time-out) parameter ℓ_v associated with each vertex v. An algorithm probes edges in E in some adaptive order within the constraint that at most ℓ_v edges are probed incident to any particular vertex v. The patience parameter can be viewed as a simple budgetary constraint, where each probe has unit cost and the patience parameter 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, then the algorithm must *commit* to the edge – that is, it must be added to the current matching. The goal is to maximize the expected size of a matching constructed in this way.

In addition to generalizing the results of Chen et al. to edge weights, Bansal et al. [6] introduced the *online bipartite stochastic matching problem.* In this problem, a single seller wishes to match their offline (indivisible) *items* to (unit-demand) *buyers* which arrive online one by one. The seller knows the possible *type/profile* of each online buyer, which is specified by edge probabilities, edge weights and a patience parameter. Here an edge probability models the likelihood a buyer type will purchase an item if the seller presents it to them, and an edge weight represents the revenue the seller will gain from making such a sale successfully. The patience of a buyer type indicates the maximum number of items they are willing to be shown. The online buyers are drawn i.i.d. from a known distribution, where the type of each online buyer is presented to the seller upon its arrival. The (potential) sale of an item to an online buyer must be made before the next online buyer arrives, and the seller's goal is to maximize their expected revenue. As in the Chen et al. model, the seller must *commit* to the first sale to which an online buyer agrees. Fata et al. observed that this problem is closely related to the multi-customer *assortment optimization problem*, which has numerous practical applications in revenue management (see [24] for details).

We study the online bipartite stochastic matching problem in the more general known i.d. setting. Specifically, each online buyer is drawn from a (potentially) distinct distribution, and the draws are done independently. When online buyers arrive adversarially, we generalize the *prophet inequality matching problem* of Alaei et al. [4]. When online buyers arrive in random order, we generalize the *prophet secretary matching problem* of Ehsani et al. [22]. We note that prophet inequalities give rise to (and in some sense are equivalent to) order oblivious posted price mechanisms, as first studied in Hajiaghayi et al. [31] and further developed for multi-parameter settings in Chawla et al. [17] and recently in Correa et al. [19]. There have been a number of very recent works studying prophet matching problems with limited distributional sample access [16, 33]. In these works, a main emphasis has been towards understanding whether a few samples is sufficient to obtain the best known competitive ratios when one is instead given full access to the distributions. Our work is motivated by analogous questions for competitive ratios in the probe-commit model, and we provide a positive answer for adversarial arrivals as well as for random order arrivals.

The online bipartite stochastic matching problem models the altruistic kidney exchange problem where the offline nodes correspond to donors and the online nodes correspond to recipients (or vice versa). A trial (probe) must be performed to determine whether a donor/recipient pair may exchange kidneys, and the edge probability corresponds to the likelihood of a permissible exchange. The online arrival setting models the restriction that the algorithm must process the recipients (or donors) in an order of which it cannot control. Our generalization of patience to downward-closed probing constraints is motivated by this application. Specifically, our framework includes knapsack/budget constraints, which allows us to model non-uniform trial costs. Another application is to online advertising, where an advertiser presents ads to consumers, and the edge probabilities represent the likelihood they will "click" on the presented ad. Basically, any matching problem in which there is uncertainty in whether the matches will succeed is a relevant application.

2 Preliminaries and Our Results

An input to the **online stochastic matching problem** with **known i.d. arrivals** firstly includes a **type graph** $H_{\text{typ}} = (U, B, F)$, which is a bipartite graph with edge weights $(w_f)_{f \in F}$ and edge probabilities $(p_f)_{f \in F}$ where $F := U \times B$. We refer to U as the **offline nodes** of H_{typ} and B as its **type nodes**. An **online probing algorithm** is given access to H_{typ}, and for each $u \in U$, and $b \in B$, $p_{u,b}$ indicates the probability that an **active** edge between u and b exists, and $w_{u,b} \geq 0$ indicates the reward for matching u to b. Given an arbitrary set S, let $S^{(*)}$ denote the set of all tuples (strings) formed from S, whose entries (characters) are all distinct. Note that we use tuple/string notation and terminology interchangeably. Each $b \in B$ has its own **(online) probing constraint** $\mathcal{C}_b \subseteq \partial(b)^{(*)}$, where $\partial(b) := U \times \{b\}$. This probing constraint indicates whether the edges of $\boldsymbol{e} = (e_1, \dots, e_k) \in \partial(b)^{(*)}$ may be probed by the algorithm in the order of its indices. Here a **probe** of an edge informs the algorithm whether or not the edge is active. We make the minimal assumption that $\mathcal{C}_b$ is **downward-closed**; that is, if $\boldsymbol{e} \in \mathcal{C}_b$, then any substring or permutation of $\boldsymbol{e}$ is also in $\mathcal{C}_b$. This includes matroid constraints, as well when $b \in B$ has a **budget** $L_b \geq 0$, and **(edge) probing costs** $(c_{u,b})_{u \in U}$, such that $\boldsymbol{e} = (e_1, \dots, e_k) \in \mathcal{C}_b$ provided $\sum_i^k c_{e_i} \leq L_b$. Observe that if b has uniform probing costs, then this corresponds to the previously discussed case of an integer **patience parameter** $\ell_b \geq 1$.

The input additionally consists of a sequence of distributions $(\mathcal{D}_i)_{i=1}^n$ supported on B, where $n \geq 1$ indicates the number of **online** vertices to be presented to the algorithm. Specifically, for $i = 1, \dots, n$, vertex v_i is drawn independently from $\mathcal{D}_i$, and we define V to be the multiset including $v_1, \dots, v_n$. The online probing algorithm executes on the **stochastic graph** $G = (U, V, E)$ where $E := U \times V$, and we denote $G \sim (H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$ to indicate G is drawn from $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$. We assume that each $e \in E$ is active independently with probability p_e, where the **edge state** $\text{st}(e) \sim \text{Ber}(p_e)$ indicates this event.

Initially, the online algorithm is only given access to $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$, yet its goal is to build a matching of active edges of G of largest possible expected weight. In the **adversarial order arrival model (AOM)**, a permutation π is generated by an **oblivious adversary**, in which case π is a function of H_{typ} and $(\mathcal{D}_i)_{i=1}^n$. In the **random order arrival model (ROM)**, π is generated u.a.r., independent of all other randomization. In either setting, π is unknown to the algorithm. For each $t = 1, \dots, n$, vertex $v_{\pi(t)}$ is presented to the algorithm, along with its edge weights, probabilities, and online probing constraint. Note that the algorithm is also presented the value $\pi(t)$, and thus learns from which distribution $v_{\pi(t)}$ was drawn. However, the edge states $(\text{st}(e))_{e \in \partial(v_{\pi(t)})}$ initially remain hidden to the algorithm. Instead, using all past available information regarding $v_{\pi(1)}, \dots, v_{\pi(t-1)}$, the algorithm must **probe** the edges of $\partial(v_{\pi(t)})$ to reveal their states, while adhering to $\mathcal{C}_{v_{\pi(t)}}$. The algorithm operates in the **probe-commit model**, in which there is a **commitment** requirement upon probing an edge. Specifically, if an edge $e = (u, v)$ is probed and turns out to be active, then the online probing algorithm must make an irrevocable decision as to whether or not to include e in its matching, prior to probing any subsequent edges. This definition of commitment is the one considered by Gupta et al. [30], and is slightly different but equivalent to the Chen et al. [18] model in which an active edge must be immediately accepted into the matching. The algorithm always has the option to pass on $v_{\pi(t)}$, yet its (potential) match must be made before the next online vertex arrives.

In general, it is easy to see that that even when the edges are unweighted and the algorithms initially knows the stochastic graph we cannot hope to obtain a non-trivial competitive ratio against the expected size of an optimal matching of the stochastic graph. Consider a stochastic graph with a single online vertex with patience 1, and $k \geq 1$ offline (unweighted) vertices where each edge e has probability $\frac{1}{k}$ of being active. The expectation of an online probing algorithm will be at most $\frac{1}{k}$ while the expected size of an optimal matching will be $1 - (1 - \frac{1}{k})^k \to 1 - \frac{1}{e}$ as $k \to \infty$. The standard approach in the literature is to instead consider the **offline stochastic matching problem** and benchmark against an *optimal offline probing algorithm* [6, 2, 14, 15]. An **offline probing algorithm** knows $G = (U, V, E)$, but initially the edge states $(\mathrm{st}(e))_{e \in E}$ are hidden. Its goal is to construct a matching of active edges of G with weight as large as possible in expectation. It can adaptively probe the edges of E in any order, but must satisfy the probing constraints $(\mathcal{C}_v)_{v \in V}$ at each step of its execution. That is, edges $\boldsymbol{e} \in E^{(*)}$ may be probed in order, provided $\boldsymbol{e}^v \in \mathcal{C}_v$ for each $v \in V$, where $\boldsymbol{e}^v$ is the substring of $\boldsymbol{e}$ restricted to edges of $\partial(v)$. It must also operate in the same probe-commit model as an online probing algorithm. We define the **(offline) adaptive benchmark** as an optimal offline probing algorithm, and denote $\mathrm{OPT}(G)$ as the expected weight of its matching when executing on G. An alternative weaker benchmark used by Brubach et al. [11, 12] is the **online adaptive benchmark**. This is defined as an optimal offline probing algorithm which executes on G and whose edge probes respect some adaptively chosen vertex ordering on V. Equivalently, the edge probes involving each $v \in V$ occur contiguously: if $e' = (u, v') \in E$ is probed after $e = (u, v)$ for $v' \neq v$, then no edge of $\partial(v)$ is probed following e'. We benchmark against $\mathbb{E}[\mathrm{OPT}(G)]$, where the expectation is over the randomness in $G \sim (H_{\mathrm{typ}}, (\mathcal{D}_i)_{i=1}^n)$. For clarity, we denote $\mathbb{E}[\mathrm{OPT}(G)]$ by $\mathrm{OPT}(H_{\mathrm{typ}}, (\mathcal{D}_i)_{i=1}^n)$.

Observe that if $p_e \in \{0, 1\}$ for each $e \in F$ of $H_{\mathrm{typ}} = (U, B, F)$, then probing is unnecessary, and the offline adaptive benchmark and the online adaptive benchmark both correspond to the expected weight of the maximum matching of G. In this special case, the online algorithm also does not need to probe edges, and so no matter which benchmark is chosen, the problem generalizes either the **prophet inequality matching problem** or the **prophet secretary matching problem**, depending on whether π is adversarial or u.a.r., respectively.

▶ **Theorem 1.** *If $\mathcal{M}(\pi)$ is the matching returned by Algorithm 8 when presented the online vertices of $G \sim (H_{typ}, (\mathcal{D}_i)_{i=1}^n)$ in an adversarial order $\pi : [n] \to [n]$, then $\mathbb{E}[w(\mathcal{M}(\pi))] \geq \frac{1}{2} OPT(H_{typ}, (\mathcal{D}_i)_{i=1}^n)$.*

▶ Remark 2. We say that Algorithm 8 attains a 1/2 competitive ratio or is 1/2-competitive (against adversarial arrivals). This is a tight bound since the problem generalizes the classic single item prophet inequality for which $\frac{1}{2}$ is an optimal competitive ratio. Recently, Brubach et al. [11, 12] independently proved the same competitive ratio against the online adaptive benchmark when G has patience values and the arrival order is adversarial yet known to the algorithm. Our results are incomparable, as their results can be applied to an *unknown* patience framework (at a loss in competitive ratio), whereas our results apply to *known* downward-closed online probing constraints, and hold against a stronger benchmark.

▶ **Theorem 3.** *If $\mathcal{M}$ is the matching returned by Algorithm 9 when presented the online vertices of $G \sim (H_{typ}, (\mathcal{D}_i)_{i=1}^n)$ in random order, then $\mathbb{E}[w(\mathcal{M})] \geq \left(1 - \frac{1}{e}\right) OPT(H_{typ}, (\mathcal{D}_i)_{i=1}^n)$.*

▶ Remark 4. In part due to its applications to multi-customer assortment optimization, the special case of identical distributions with one-sided patience values has been studied in multiple works [6, 2, 14, 15], beginning with the 0.12 competitive ratio of Bansal et al. [6]. The previously best known competitive ratio of 0.46 for arbitrary patiences is due to Brubach et al. [15]. Fata et al. [24] improved this competitive ratio to 0.51 for the special case of **unbounded patience**. In an early 2020 arXiv version of this paper [8], we proved a competitive ratio of $1-1/e$ for arbitrary patience values. All these previous competitive ratios (including ours) are proven against the offline adaptive benchmark. Theorem 3 generalizes this result, as it is the first to apply to non-identical distributions, as well as to more general probing constraints. Brubach et al. [11, 12] independently achieved a $1 - 1/e$ competitive ratio for arbitrary patience values in the known i.i.d. setting, however their ratio is against the weaker online adaptive benchmark, and so is incomparable with previous results in the literature. Interestingly, $1 - 1/e$ remains the best known competitive ratio in the prophet secretary matching problem due to Ehsani et al. [22], despite significant progress in the case of a single offline node (see [5, 20]). Huang et al. [32] very recently proved a 0.703 hardness result for multiple offline nodes and known i.i.d. arrivals.

In order to discuss the efficiency of our algorithms in the generality of our probing constraints, we work in the **membership oracle model**. An online probing algorithm may make a **membership query** to any string $\boldsymbol{e} \in \partial(b)^{(*)}$ for $b \in B$, thus determining in a single operation whether or not $\boldsymbol{e} \in \partial(b)^{(*)}$ is in $\mathcal{C}_b$. All our algorithms are implementable in polynomial time, as we prove in the full version of the paper (hereby denoted [9]).

A well studied special case of the online stochastic matching problem with known i.d. online arrivals is the case of a **known stochastic graph** (see [18, 1, 6, 2, 7, 27, 13, 35]). In this setting, the input $H_{\text{typ}} = (V, B, F)$ satisfies $n = |B|$, and the distributions $(\mathcal{D}_i)_{i=1}^n$ are all point-mass on *distinct* vertices of B. Thus, the online vertices of G are not randomly drawn, and G is instead equal to H_{typ}. The online probing algorithm thus knows the stochastic graph G in advance, but remains unaware of the edge states $(\text{st}(e))_{e \in E}$, and so it still must sequentially probe the edges to reveal their states. Again, it must operate in the probe-commit model, and respect the probing constraints $(\mathcal{C}_v)_{v \in V}$ as well as the arrival order π on V.

▶ **Corollary 5** (of Theorem 3)**.** *If $\mathcal{M}$ is the matching returned by Algorithm 6 when presented the online vertices of G in random order, then $\mathbb{E}[w(\mathcal{M})] \geq \left(1 - \frac{1}{e}\right) OPT(G)$.*

▶ Remark 6. When Algorithm 9 executes in the known graph setting, it is **non-adaptive** in that its probes are a (randomized) function of G. In [9], we complement Corollary 5 with a $1 - 1/e$ hardness result which applies to *all* non-adaptive probing algorithms (even probing algorithms which execute offline, and thus do not respect the arrival order π of V).

▶ Remark 7. Gamlath et al. [27] consider an online probing algorithm when G is **unconstrained** – i.e., $\mathcal{C}_v = \partial(v)^{(*)}$ for all $v \in V$ – and known to the algorithm. Both our algorithm and theirs attain a performance guarantee of $1 - 1/e$ against very different non-standard LPs – LP-config and LP-QC, respectively. Note that LP-QC has exponentially many constraints and polynomially many variables, whereas LP-config has polynomially many constraints and exponentially many variables (see Appendix B for a statement of LP-QC). To the best of our knowledge, LP-QC does not seem to have an extension even to arbitrary patience values, as it is unclear how to generalize its constraints while maintaining polynomial time solvability. Despite having such different forms, in the unconstrained setting the LPs take on the same value, as we prove in Proposition 28 of Appendix B. Thus, Theorem 3 can be viewed as a generalization of their work to downward-closed online probing constraints and known i.d. random order arrivals. Very recently, Pollner et al. [35] proved a 0.426 competitive ratio against the offline adaptive benchmark in the special case of a bipartite graph with (one-sided) patience values. Our results are incomparable, as their algorithm works for random order *edge arrivals*, whereas ours requires one-sided random order vertex arrivals, yet has a better competitive ratio and works for more general probing constraints.

2.1 An Overview of Our Techniques

For simplicity, we first describe our techniques in the known stochastic graph setting. Afterwards, we explain how our techniques extend to the known i.d. setting. Let us suppose that we are presented a stochastic graph $G = (U, V, E)$. For the case of patience values $(\ell_v)_{v \in V}$, a natural solution is to solve an LP introduced by Bansal et al. [6] to obtain fractional values for the edges of G, say $(x_e)_{e \in E}$, such that x_e upper bounds the probability e is probed by the offline adaptive benchmark. Clearly, $\sum_{e \in \partial(v)} x_e \leq \ell_v$ is a constraint for each $v \in V$, and so by applying a dependent rounding algorithm (such as the GKSP algorithm of Gandhi et al. [28]), one can round the values $(x_e)_{e \in \partial(v)}$ to determine ℓ_v edges of $\partial(v)$ to probe. By probing these edges in a carefully chosen order, and matching v to the first edge revealed to be active, one can guarantee that each $e \in \partial(v)$ is matched with probability reasonably close to $p_e x_e$. This is the high-level approach used in many stochastic matching algorithms (for example [6, 2, 7, 15, 13, 35]). However, even for a single online node, this LP overestimates the value of the offline adaptive benchmark, and so any algorithm designed in this way will match certain edges with probability strictly less than $p_e x_e$. This is problematic, for the value of the match made to v is ultimately compared to $\sum_{e \in \partial(v)} p_e w_e x_e$, the contribution of the variables $(x_e)_{e \in \partial(v)}$ to the LP solution. In fact, Fata et al. [24] showed that the ratio between $\mathrm{OPT}(G)$ and an optimum solution to this LP can be as small as 0.51, so the $1 - 1/e$ competitive ratio of Theorem 3 cannot be achieved via a comparison to this LP, even for the special case of patience values.

Defining LP-config. Our approach is to work with a configuration LP (LP-config) which we initially called LP-new in our 2020 arXiv paper [8] and used in our companion paper [10] to attain an (optimal) $1/e$ competitive ratio for the edge-weighted secretary matching problem in the probe-commit model. This LP has exponentially many variables which accounts for the many probing strategies available to an arriving vertex v with probing constraint $\mathcal{C}_v$. For each $\boldsymbol{e} \in E^{(*)}$, define $q(\boldsymbol{e}) = \prod_{f \in \boldsymbol{e}} (1 - p_f)$, to be the probability that all the edges of $\boldsymbol{e}$ are inactive, where $q(\lambda) := 1$ for the empty string/character λ. For $f \in \boldsymbol{e}$, we denote $\boldsymbol{e}_{<f}$ to be the substring of $\boldsymbol{e}$ from its first edge up to, but not including, f. Observe then that $\mathrm{val}(\boldsymbol{e}) := \sum_{f \in \boldsymbol{e}} w_f \cdot p_f \cdot q(\boldsymbol{e}_{<f})$ corresponds to the expected weight of the first active edge revealed if $\boldsymbol{e}$ is probed in order of its entries. For each $v \in V$, we introduce a decision variable $x_v(\boldsymbol{e})$ and write the following LP:

$$\text{maximize} \quad \sum_{v \in V} \sum_{\boldsymbol{e} \in \mathcal{C}_v} \mathrm{val}(\boldsymbol{e}) \cdot x_v(\boldsymbol{e}) \tag{LP-config}$$

$$\text{subject to} \quad \sum_{v \in V} \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_v: \\ (u,v) \in \boldsymbol{e}}} p_{u,v} \cdot q(\boldsymbol{e}_{<(u,v)}) \cdot x_v(\boldsymbol{e}) \leq 1 \qquad \forall u \in U \tag{1}$$

$$\sum_{\boldsymbol{e} \in \mathcal{C}_v} x_v(\boldsymbol{e}) = 1 \qquad \forall v \in V, \tag{2}$$

$$x_v(\boldsymbol{e}) \geq 0 \qquad \forall v \in V, \boldsymbol{e} \in \mathcal{C}_v \tag{3}$$

In this work, we provide the first proof that LP-config is a **relaxation** of the offline adaptive benchmark. This result is stated but not proven in our companion paper, instead crediting the full arXiv version [9] of this paper. Unlike previous LPs used in the literature, we are not aware of an easy proof of this fact, and so we consider our proof to be a technical contribution.

▶ **Theorem 8.** *$OPT(G) \leq LPOPT(G)$.*

▶ Remark 9. For the case of patience values, a closely related LP was independently introduced by Brubach et al. [11, 12] to design probing algorithms for known i.i.d. arrivals and known i.d. adversarial arrivals. Their competitive ratios are proven against an optimal solution to this LP, which they argue relaxes the *online* adaptive benchmark.

When each $\mathcal{C}_v$ is downward-closed, LP-config can be solved efficiently by using a deterministic separation oracle for the dual of LP-config, in conjunction with the ellipsoid algorithm [36, 29]. In [10], we introduce a greedy probing algorithm for offline vertex weights which attains $1/2$ and $1 - 1/e$ competitive ratios for adversarial and random order arrivals, respectively. These ratios are proven by applying the primal-dual method to a non-standard LP (distinct from LP-config). We also showed that this greedy probing algorithm can be used as a separation oracle for the dual of LP-config, as this ensures our $1/e$-competitive edge weights algorithm is efficient. For completeness, we provide the details for extending to the known i.d. case in [9], as well as a buyer/seller interpretation of the separation oracle problem.

Proving Theorem 8. In order to prove Theorem 8, the natural approach is to view $x_v(\boldsymbol{e})$ as the probability that the offline adaptive benchmark probes the edges of $\boldsymbol{e}$ in order, where $v \in V$ and $\boldsymbol{e} \in \mathcal{C}_v$. Let us suppose that hypothetically we could make the following restrictive assumptions regarding the offline adaptive benchmark:

P_1 If $e = (u, v)$ is probed and $\text{st}(e) = 1$, then e is included in the matching, provided v is currently unmatched.

P_2 For each $v \in V$, the edge probes involving $\partial(v)$ are made independently of the edge states $(\text{st}(e))_{e \in \partial(v)}$.

Observe then that P_1 and P_2 would imply that the expected weight of the edge assigned to v is $\sum_{\boldsymbol{e} \in \mathcal{C}_v} \text{val}(\boldsymbol{e}) \cdot x_v(\boldsymbol{e})$. Moreover, the left-hand side of (1) would correspond to the probability $u \in U$ is matched, so $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$ would be a feasible solution to LP-config, and so we could upper bound $\text{OPT}(G)$ by $\text{LPOPT}(G)$. Now, if we were working with the *online* adaptive benchmark, then it is clear that we could assume P_1 and P_2 simultaneously[1] w.l.o.g. On the other hand, if a probing algorithm does *not* respect an adaptive vertex ordering on V (i.e,, does not probe edges in $\partial(v)$ consecutively), then the probes involving $v \in V$ will in general depend on $(\text{st}(e))_{e \in \partial(v)}$. For instance, if $e \in \partial(v)$ is probed and inactive, then perhaps the offline adaptive benchmark next probes $e' = (u, v') \in \partial(v')$ for some $v' \neq v$. If e' is active and thus added to the matching by P_1, then the offline adaptive benchmark can never subsequently probe (u, v) without violating P_1, as u is now unavailable to be matched to v. Thus, the natural interpretation of the decision variables of LP-config does not seem to easily lend itself to a proof of Theorem 8.

Our solution is to consider a **combinatorial relaxation** of the offline stochastic matching problem, which we define to be a new stochastic probing problem on G whose optimal value $\text{OPT}_{\text{rel}}(G)$ satisfies $\text{OPT}(G) \leq \text{OPT}_{\text{rel}}(G)$. We refer to this problem as the **relaxed**

[1] It is clear that we may assume the offline adaptive benchmark satisfies P_1 w.l.o.g., but not P_2.

stochastic matching problem, a solution to which is a **relaxed probing algorithm**. Roughly speaking, a relaxed probing algorithm operates in the same framework as an offline probing algorithm, yet it returns a one-sided matching of the online vertices which matches each offline node at most once *in expectation*. We provide a precise definition in Section 3. Crucially, there exists an *optimal* relaxed probing algorithm which is **non-adaptive** – that is, a (randomized) function of G – and which satisfies P_1. Non-adaptivity is a much stronger property than P_2, and so by the above discussion we are able to conclude that $\mathrm{OPT}_{\mathrm{rel}}(G) \leq \mathrm{LPOPT}(G)$. Since $\mathrm{OPT}(G) \leq \mathrm{OPT}_{\mathrm{rel}}(G)$ by construction, this implies Theorem 8. Proving the existence of an optimal relaxed probing algorithm which is non-adaptive is one of the most technically challenging parts of the paper, and is the main content of Lemma 14 of Section 3. Note that there may be a simpler proof of Theorem 8, however our relaxed stochastic matching problem exactly characterizes LP-config (i.e., $\mathrm{OPT}_{\mathrm{rel}}(G) = \mathrm{LPOPT}(G)$), and so it helps us understand LP-config. For instance, in Appendix B, we show that in the unconstrained patience setting, LP-QC of [27] is also characterized by our relaxed matching problem. This implies that the LPs take on the same value, despite having very different formulations in this special setting.

Defining the probing algorithms: After proving that LP-config is a relaxation of the offline adaptive benchmark, we use it to design online probing algorithms. Suppose that we are presented a feasible solution, say $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$, to LP-config for G. For each $e \in E$, define

$$\widetilde{x}_e := \sum_{\substack{\boldsymbol{e}' \in \mathcal{C}_v: \\ e \in \boldsymbol{e}'}} q(\boldsymbol{e}'_{<e}) \cdot x_v(\boldsymbol{e}'). \tag{4}$$

We refer to the values $(\widetilde{x}_e)_{e \in E}$ as the **edge variables** of the solution $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$. If we now fix $s \in V$, then we can easily leverage constraint (2) to design a simple *fixed vertex* probing algorithm which matches each edge of $e \in \partial(s)$ with probability *exactly* equal to $p_e \widetilde{x}_e$. Specifically, draw $\boldsymbol{e}' \in \mathcal{C}_s$ with probability $x_s(\boldsymbol{e}')$. If $\boldsymbol{e}' = \lambda$, then return the empty set. Otherwise, set $\boldsymbol{e}' = (e'_1, \ldots, e'_k)$ for $k := |\boldsymbol{e}'| \geq 1$, and probe the edges of $\boldsymbol{e}'$ in order. Return the first edge which is revealed to be active, if such an edge exists. Otherwise, return the empty set. We refer to this algorithm as `VertexProbe`, and denote its output on the input $(s, \partial(s), (x_s(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_s})$ by $\texttt{VertexProbe}(s, \partial(s), (x_s(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_s})$.

▶ **Lemma 10.** *For each $e \in \partial(s)$, $\mathbb{P}[\texttt{VertexProbe}(s, \partial(s), (x_s(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_s}) = e] = p_e \widetilde{x}_e$.*

▶ Remark 11. We can view Lemma 10 as an **exact rounding guarantee**. The fact that such a guarantee exists, no matter the choice of $\mathcal{C}_s$, is one of the main benefits of working with LP-config, opposed to LP-std or LP-QC. As discussed, a solution to LP-std provably cannot be rounded exactly in this way. There *does* exist an exact rounding guarantee for LP-QC, however it only applies to the unconstrained setting of $\mathcal{C}_v = \partial(s)^{(*)}$, and the procedure is much more complicated than ours (see Theorem 29 of Appendix B for details).

▶ **Definition 12.** *We say that* `VertexProbe` ***commits*** *to the edge $e = (u, s) \in \partial(s)$, or equivalently the vertex $u \in N(s)$, provided the algorithm outputs e when executing on the fixed node $s \in V$. When it is clear that* `VertexProbe` *is being executed on s, we say that s commits to e (equivalently the vertex u).*

Consider now the following online probing algorithm, where π is either u.a.r. or adversarial.

Algorithm 1 Known Stochastic Graph.

Require: a stochastic graph $G = (U, V, E)$.
Ensure: a matching $\mathcal{M}$ of active edges of G.
1: $\mathcal{M} \leftarrow \emptyset$.
2: Compute an optimal solution of LP-config for G, say $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$
3: **for** $s \in V$ in order based on π **do**
4: Set $e \leftarrow$ `VertexProbe`$(s, \partial(s), (x_s(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_s})$.
5: **if** $e = (u, s)$ for some $u \in U$, and u is unmatched **then** ▷ this line ensures $e \neq \emptyset$
6: Add e to $\mathcal{M}$.
7: **end if**
8: **end for**
9: **return** $\mathcal{M}$.

▶ Remark 13. Technically, line (6) should occur within the `VertexProbe` subroutine to adhere to the probe-commit model, however we express our algorithms in this way for conciseness.

Improvement via online contention resolution. Algorithm 1 does not attain a constant competitive ratio for adverarial arrivals, and its competitive ratio is only 1/2 in the random order arrivals. Thus, we must modify the algorithm to prove Theorems 1 and 3, even in the known stochastic graph setting. Our modification involves concurrently applying an appropriate rank one matroid **contention resolution scheme (CRS)** to each offline vertex of G, a concept formalized much more generally in the seminal paper by Chekuri, Vondrak, and Zenklusen [38]. Contention resolution has become a fundamental tool for stochastic optimization problems, and we illustrate its versatility by applying it to a non-standard LP.

Fix $u \in U$, and observe that constraint (1) ensures that $\sum_{e \in \partial(u)} p_e \widetilde{x}_e \leq 1$. Moreover, if we set $z_e := p_e \widetilde{x}_e$, then observe that as `VertexProbe` executes on v, each edge $e = (u, v) \in \partial(u)$ is committed to u independently with probability z_e. On the other hand, there may be many edges which commit to u so we must resolve which one to take. In Algorithm 1, u is matched greedily to the first online vertex which commits to it, regardless of how π is generated. We apply existing **online** and **random order** contention resolution schemes to ensure that e is matched to u with probability $1/2 \cdot z_e$ when π is generated by an adversary, and $(1 - 1/e) \cdot z_e$ when π is generated u.a.r. These lower bounds on the edge variables allow us to conclude the desired competitive ratios, as $\sum_{e \in E} w_e p_e \widetilde{x}_e$ upper bounds $\text{OPT}(G)$ by Theorem 8. We provide the specific schemes used for adversarial arrivals and random orders arrivals in Section 4. In the latter setting, the CRS based approach simplifies the pricing based approach Gamlath et al. [27] used to attain a competitive ratio of $1 - 1/e$ in the special unconstrained setting (see Remark 7). This simplified approach was also observed by Fu et al. [26] in the context of the Gamlath et al. LP (LP-QC). They focus on the unconstrained probe-commit model, and design a 8/15-competitive algorithm for general graph random order vertex arrivals. It remains open whether their results can be extended to general patience values and random order edge arrivals. For context, 0.395 is the best known competitive ratio when allowing for arbitrary patience values and random order edge arrivals [35]. We focus on the bipartite graphs with one-sided arrivals, as the main goal of this paper was to fully resolve the complications posed by one-sided probing constraints in this arrival model.

Extending to known i.d. arrivals. In Appendix A, we prove Theorems 1 and 3 in their full generality when G is unknown and drawn from $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$. We do so by first generalizing LP-config to a new LP called LP-config-id. This LP departs from previous ones used in the probing literature, as it depends both on the type graph as well as the distributions. For each $i \in [n]$, we introduce a collection of variables $(x_i(\boldsymbol{e} \,||\, b))_{\boldsymbol{e} \in \mathcal{C}_b, b \in B}$ associated with the distribution $\mathcal{D}_i$. We again apply known contention resolution schemes, however the additional variables associated with the possible types of $v_i \sim \mathcal{D}_i$ introduce correlated events which must be treated delicately in the context of **CRS selectibility**. Crucially, the schemes we employ do *not* make use of the type of vertex v_i, and so we are able to argue that analogous edge variable lower bounds hold as in the known stochastic graph setting.

3 Relaxing the Offline Adaptive Benchmark via LP-config

Given a stochastic graph $G = (U, V, E)$, we define the **relaxed stochastic matching problem**. A solution to this problem is a **relaxed probing algorithm** $\mathcal{A}$, which operates in the previously described framework of an (offline) probing algorithm. That is, $\mathcal{A}$ is firstly given access to a stochastic graph $G = (U, V, E)$. Initially, the edge states $(\text{st}(e))_{e \in E}$ are unknown to $\mathcal{A}$, and $\mathcal{A}$ must adaptivity probe these edges to reveal their states, while respecting the downward-closed probing constraints $(\mathcal{C}_v)_{v \in V}$. As in the offline problem, $\mathcal{A}$ returns a subset $\mathcal{M}$ of its active edge probes, and its goal is to maximize $\mathbb{E}[w(\mathcal{M})]$, where $w(\mathcal{M}) := \sum_{e \in \mathcal{M}} w_e$. However, unlike before where $\mathcal{M}$ was required to be a matching of G, we relax the required properties of $\mathcal{M}$:

1. Each $v \in V$ appears in at most one edge of $\mathcal{M}$.
2. If N_u counts the number of edges of $\partial(u)$ which are included in $\mathcal{M}$, then $\mathbb{E}[N_u] \leq 1$ for each $u \in U$.

We refer to $\mathcal{M}$ as a **one-sided matching** of the online nodes, and abuse terminology slightly and say that $e \in E$ is matched by $\mathcal{A}$ if $e \in \mathcal{M}$. In constructing $\mathcal{M}$, $\mathcal{A}$ must operate in the previously described probe-commit model. We define the **relaxed benchmark** as an optimal relaxed probing algorithm, and denote its expected value when executing on G by $\text{OPT}_{\text{rel}}(G)$. Observe that since any offline probing algorithm is a relaxed probing algorithm, we have that

$$\text{OPT}(G) \leq \text{OPT}_{\text{rel}}(G). \tag{5}$$

We say that $\mathcal{A}$ is **non-adaptive**, provided the probes are a (randomized) function of G. Equivalently, $\mathcal{A}$ is non-adaptive if the probes of $\mathcal{A}$ are statistically independent from $(\text{st}(e))_{e \in E}$. Unlike for the offline stochastic matching problem, there exists a relaxed probing algorithm which is both optimal *and* non-adaptive:

▶ **Lemma 14.** *For any stochastic graph $G = (U, V, E)$ with downward-closed probing constraints $(\mathcal{C}_v)_{v \in V}$, there exists an optimum relaxed probing algorithm $\mathcal{B}$ which satisfies the following properties:*

Q_1 *If $e = (u, v)$ is probed, $st(e) = 1$, and v was previously unmatched, then $\mathcal{B}$ matches e.*

Q_2 *$\mathcal{B}$ is non-adaptive on G.*

▶ Remark 15. Note that Q_2 implies the hypothetical property P_2, yet is much stronger.

Let us assume that Lemma 14 holds for now.

Proof of Theorem 8. Consider $\mathcal{B}$ of Lemma 14, and define $x_v(\boldsymbol{e})$ to be the probability that $\mathcal{B}$ probes the edges of $\boldsymbol{e}$ in order for $v \in V$ and $\boldsymbol{e} \in \mathcal{C}_v$. Since $\mathcal{B}$ is a relaxed probing algorithm, we can apply properties Q_1 and Q_2 to show that $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$ is a feasible solution to LP-config. Moreover, if $\mathcal{N}$ is returned when $\mathcal{B}$ executes on G, then

$$\mathbb{E}[w(\mathcal{N})] = \sum_{v \in V} \sum_{\boldsymbol{e} \in \mathcal{C}_v} \text{val}(\boldsymbol{e}) \cdot x_v(\boldsymbol{e}).$$

Thus, the optimality of $\mathcal{B}$ implies that $\text{OPT}_{\text{rel}}(G) \leq \text{LPOPT}(G)$, and so together with (5), Theorem 8 follows. ◀

▶ Remark 16. As mentioned, LP-config is an exact LP formulation of the relaxed stochastic matching problem, as we prove in Theorem 27 of Appendix B.

3.1 Proving Lemma 14

Let us suppose that $G = (U, V, E)$ is a stochastic graph with downward-closed probing constraints $(\mathcal{C}_v)_{v \in V}$. In order to prove Lemma 14, we must show that there exists an optimal relaxed probing algorithm which is non-adaptive and satisfies Q_1. Our high level approach is to consider an optimal relaxed probing algorithm $\mathcal{A}$ which satisfies Q_1, and then to construct a new non-adaptive algorithm $\mathcal{B}$ by *stealing* the strategy of $\mathcal{A}$, without any loss in performance. More specifically, we construct $\mathcal{B}$ by writing down for each $v \in V$ and $\boldsymbol{e} \in \mathcal{C}_v$ the probability that $\mathcal{A}$ probes the edges of $\boldsymbol{e}$ in order. These probabilities necessarily satisfy certain inequalities which we make use of in designing $\mathcal{B}$. In order to do so, we need a technical randomized rounding procedure whose precise relevance will become clear in the proof of Lemma 14.

Suppose that $\boldsymbol{e} \in E^{(*)}$, and recall that λ is the empty string/character. Let us now assume that $(y_v(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_v}$ is a collection of non-negative values which satisfy $y_v(\lambda) = 1$, and

$$\sum_{\substack{e \in \partial(v): \\ (\boldsymbol{e}', e) \in \mathcal{C}_v}} y_v(\boldsymbol{e}', e) \leq y_v(\boldsymbol{e}'), \tag{6}$$

for each $\boldsymbol{e}' \in \mathcal{C}_v$. For space considerations, we defer the proof of the below proposition to [9].

▶ **Proposition 17.** *Given a collection of values $(y_v(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_v}$ which satisfy $y_v(\lambda) = 1$ and (6), there exists a distribution $\mathcal{D}^v$ supported on $\mathcal{C}_v$, such that if $\boldsymbol{Y} \sim \mathcal{D}^v$, then for each $\boldsymbol{e} = (e_1, \ldots, e_k) \in \mathcal{C}_v$ with $k := |\boldsymbol{e}| \geq 1$, it holds that*

$$\mathbb{P}[(\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_k) = (e_1, \ldots, e_k)] = y_v(\boldsymbol{e}), \tag{7}$$

where $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_k$ are the first k characters of $\boldsymbol{Y}$ (where $\boldsymbol{Y}_i := \lambda$ if $\boldsymbol{Y}$ has no i^{th} character).

Proof of Lemma 14. Suppose that $\mathcal{A}$ is an optimal relaxed probing algorithm which returns the one-sided matching $\mathcal{M}$ after executing on the stochastic graph $G = (U, V, E)$. In a slight abuse of terminology, we say that e is matched by $\mathcal{A}$, provided e is included in $\mathcal{M}$. We shall also make the simplifying assumption that $p_e < 1$ for each $e \in E$, as the proof can be clearly adapted to handle the case when certain edges have $p_e = 1$ by restricting which strings of each $\mathcal{C}_v$ are considered.

Observe that since $\mathcal{A}$ is optimal, it is clear that we may assume the following properties hold w.l.o.g. for each $e \in E$:

1. e is probed only if e can be added to the currently constructed one-sided matching.
2. If e is probed and $\mathrm{st}(e) = 1$, then e is included in $\mathcal{M}$.

Thus, in order to prove the lemma, we must find an alternative algorithm $\mathcal{B}$ which is non-adaptive, yet continues to be optimal. To this end, we shall first express $\mathbb{E}[w(\mathcal{M}(v))]$ in a convenient form for each $v \in V$, where $w(\mathcal{M}(v))$ is the weight of the edge matched to v (which is 0 if no match occurs).

Given $v \in V$ and $1 \leq i \leq |U|$, we define X_i^v to be the i^{th} edge adjacent to v that is probed by $\mathcal{A}$. This is set equal to λ by convention, provided no such edge exists. We may then define $\boldsymbol{X}^v := (X_1^v, \ldots, X_{|U|}^v)$, and $\boldsymbol{X}_{\leq k}^v := (X_1^v, \ldots, X_k^v)$ for each $1 \leq k \leq |U|$. Moreover, given $\boldsymbol{e} = (e_1, \ldots, e_k) \in E^{(*)}$ with $k \geq 1$, define $S(\boldsymbol{e})$ to be the event in which e_k is the only active edge amongst $e_1, \ldots, e_k$. Observe then that

$$\mathbb{E}[w(\mathcal{M}(v))] = \sum_{\substack{\boldsymbol{e}=(e_1,\ldots,e_k)\in\mathcal{C}_v:\\ k\geq 1}} w_{e_k}\mathbb{P}[S(\boldsymbol{e})\cap\{\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}\}],$$

as (1) and (2) ensure v is matched to the first probed edge which is revealed to be active. Moreover, if $\boldsymbol{e}=(e_1,\ldots,e_k)\in\mathcal{C}_v$ for $k\geq 2$, then

$$\mathbb{P}[S(\boldsymbol{e})\cap\{\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}\}] = \mathbb{P}[\{\mathrm{st}(e_k)=1\}\cap\{\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}\}], \tag{8}$$

as (1) and (2) ensure $\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}$ only if $e_1,\ldots,e_{k-1}$ are inactive. Thus,

$$\begin{aligned}
\mathbb{E}[w(\mathcal{M}(v))] &= \sum_{\substack{\boldsymbol{e}=(e_1,\ldots,e_k)\in\mathcal{C}_v:\\ k\geq 1}} w_{e_k}\mathbb{P}[S(\boldsymbol{e})\cap\{\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}\}]\\
&= \sum_{\substack{\boldsymbol{e}=(e_1,\ldots,e_k)\in\mathcal{C}_v:\\ k\geq 1}} w_{e_k}\mathbb{P}[\{\mathrm{st}(e_k)=1\}\cap\{\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}\}]\\
&= \sum_{\substack{\boldsymbol{e}=(e_1,\ldots,e_k)\in\mathcal{C}_v:\\ k\geq 1}} w_{e_k}p_{e_k}\mathbb{P}[\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}],
\end{aligned}$$

where the final equality holds since $\mathcal{A}$ must decide on whether to probe e_k prior to revealing $\mathrm{st}(e_k)$. As a result, after summing over $v\in V$,

$$\mathbb{E}[w(\mathcal{M})] = \sum_{v\in V}\sum_{\substack{\boldsymbol{e}=(e_1,\ldots,e_k)\in\mathcal{C}_v:\\ k\geq 1}} w_{e_k}p_{e_k}\mathbb{P}[\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}]. \tag{9}$$

Our goal is to find a non-adaptive relaxed probing algorithm which matches the value of (9). Thus, for each $v\in V$ and $\boldsymbol{e}=(e_1,\ldots,e_k)\in\mathcal{C}_v$ with $k\geq 1$, define $x_v(\boldsymbol{e}) := \mathbb{P}[\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}]$, where $x_v(\lambda):=1$. Observe now that for each $\boldsymbol{e}'=(e'_1,\ldots,e'_k)\in\mathcal{C}_v$,

$$\sum_{\substack{e\in\partial(v):\\ (\boldsymbol{e}',e)\in\mathcal{C}_v}} \mathbb{P}[\boldsymbol{X}^v_{\leq k+1}=(\boldsymbol{e}',e)\mid \boldsymbol{X}^v_{\leq k}=\boldsymbol{e}'] \leq 1-p_{e'_k}. \tag{10}$$

To see (10), observe that the the left-hand side corresponds to the probability $\mathcal{A}$ probes some edge $e\in\partial(v)$, given it already probed $\boldsymbol{e}'$ in order. On the other hand, if a subsequent edge is probed, then (1) and (2) imply that e'_k must have been inactive, which occurs independently of the event $\boldsymbol{X}^v_{\leq k}=\boldsymbol{e}'$. This explains the right-hand side of (10). Using (10), the values $(x_v(\boldsymbol{e}))_{\boldsymbol{e}\in\mathcal{C}_v}$ satisfy

$$\sum_{\substack{e\in\partial(v):\\ (\boldsymbol{e}',e)\in\mathcal{C}_v}} x_v(\boldsymbol{e}',e) \leq (1-p_{e'_k})\cdot x_v(\boldsymbol{e}'), \tag{11}$$

for each $\boldsymbol{e}'=(e'_1,\ldots,e'_k)\in\mathcal{C}_v$ with $k\geq 1$. Moreover, clearly $\sum_{e\in\partial(v)} x_v(e)\leq 1$.

Given $\boldsymbol{e}=(e_1,\ldots,e_k)\in\mathcal{C}_v$ for $k\geq 1$, recall that $\boldsymbol{e}_{<k}:=(e_1,\ldots,e_{k-1})$ where $\boldsymbol{e}_{<1}:=\lambda$ if $k=1$. Moreover, $q(\boldsymbol{e}_{<k}):=\prod_{i=1}^{k-1}(1-p_{e_i})$, where $q(\lambda):=1$. Using this notation, define for each $\boldsymbol{e}\in\mathcal{C}_v$

$$y_v(\boldsymbol{e}) := \begin{cases} x_v(\boldsymbol{e})/q(\boldsymbol{e}_{<|\boldsymbol{e}|}) & \text{if } |\boldsymbol{e}|\geq 1,\\ 1 & \text{otherwise.}\end{cases} \tag{12}$$

Observe that (11) ensures that for each $\boldsymbol{e}' \in \mathcal{C}_v$,

$$\sum_{\substack{e \in \partial(v): \\ (\boldsymbol{e}', e) \in \mathcal{C}_v}} y_v(\boldsymbol{e}', e) \leq y_v(\boldsymbol{e}'), \tag{13}$$

and $y_v(\lambda) := 1$. As a result, Proposition 17 implies that for each $v \in V$, there exists a distribution $\mathcal{D}^v$ such that if $\boldsymbol{Y}^v \sim \mathcal{D}^v$, then for each $\boldsymbol{e} \in \mathcal{C}_v$ with $|\boldsymbol{e}| = k \geq 1$,

$$\mathbb{P}[\boldsymbol{Y}^v_{\leq k} = \boldsymbol{e}] = y_v(\boldsymbol{e}). \tag{14}$$

Moreover, $\boldsymbol{Y}^v$ is drawn independently from the edge states, $(\mathrm{st}(e))_{e \in E}$. Consider now the following algorithm $\mathcal{B}$, which satisfies the desired properties Q_1 and Q_2 of Lemma 14:

Algorithm 2 Algorithm $\mathcal{B}$.

Require: a stochastic graph $G = (U, V, E)$.
Ensure: a one-sided matching $\mathcal{N}$ of G of active edges.

Set $\mathcal{N} \leftarrow \emptyset$.
Draw $(\boldsymbol{Y}^v)_{v \in V}$ according to the product distribution $\prod_{v \in V} \mathcal{D}^v$.
for $v \in V$ **do**
  for $i = 1, \ldots, |\boldsymbol{Y}^v|$ **do**
    Set $e \leftarrow \boldsymbol{Y}^v_i$. ▷ $\boldsymbol{Y}^v_i$ is the i^{th} edge of $\boldsymbol{Y}^v$
    Probe the edge e, revealing $\mathrm{st}(e)$.
    if $\mathrm{st}(e) = 1$ and v is unmatched by $\mathcal{N}$ **then**
      Add e to $\mathcal{N}$.
    end if
  end for
end for
return $\mathcal{N}$.

Using (14) and the non-adaptivity of $\mathcal{B}$, it is clear that for each $v \in V$,

$$\begin{aligned}
\mathbb{E}[w(\mathcal{N}(v))] &= \sum_{\substack{\boldsymbol{e} = (e_1, \ldots, e_k) \in \mathcal{C}_v: \\ k \geq 1}} w_{e_k} \mathbb{P}[S(\boldsymbol{e})] \cdot \mathbb{P}[\boldsymbol{Y}^v_{\leq k} = \boldsymbol{e}] \\
&= \sum_{\substack{\boldsymbol{e} = (e_1, \ldots, e_k) \in \mathcal{C}_v: \\ k \geq 1}} w_{e_k} p_{e_k} q(\boldsymbol{e}_{<k}) y_v(\boldsymbol{e}) \\
&= \sum_{\substack{\boldsymbol{e} = (e_1, \ldots, e_k) \in \mathcal{C}_v: \\ k \geq 1}} w_{e_k} p_{e_k} x_v(\boldsymbol{e}) = \mathbb{E}[w(\mathcal{M}(v))].
\end{aligned}$$

Thus, after summing over $v \in V$, it holds that $\mathbb{E}[w(\mathcal{N})] = \mathbb{E}[w(\mathcal{M})] = \mathrm{OPT}_{\mathrm{rel}}(G)$, and so in addition to satisfying Q_1 and Q_2, $\mathcal{B}$ is optimal. Finally, it is easy to show that each $u \in U$ is matched by $\mathcal{N}$ at most once in expectation since $\mathcal{M}$ has this property. Thus, $\mathcal{B}$ is a relaxed probing algorithm which is optimal and satisfies the required properties of Lemma 14. ◀

4 Proving Theorems 1 and 3 for a Known Stochastic Graph

Given $k \geq 1$, consider the ground set $[k] := \{1, \ldots, k\}$, and $\mathcal{P} := \{\boldsymbol{z} \in [0,1]^k : \sum_{i=1}^k z_i \leq 1\}$. Fix $\boldsymbol{z} \in \mathcal{P}$, and let $R(\boldsymbol{z}) \subseteq [k]$ denote the random set where each $i \in [k]$ is included in $R(\boldsymbol{z})$ independently with probability z_i. Feldman et al. [25] considered a restricted class of contention resolution schemes called **online contention resolution schemes** (OCRS). The elements of $[k]$ are presented to the OCRS ψ in adversarial order, where in each step, an arriving $i \in [k]$ reveals if it is in $R(\boldsymbol{z})$, at which point ψ must make an irrevocable decision as to whether it wishes to return i as its output. We refer the reader to [9] for a brief overview of CRS terminology.

Suppose the elements of $[k]$ arrive according to some permutation $\sigma : [k] \to [k]$ (i.e., $\sigma(1), \ldots, \sigma(k)$), and $\boldsymbol{z} \in [0,1]^k$ satisfies $\sum_{i=1}^{k} z_i \leq 1$. Upon the arrival of element $\sigma(t) \in [k]$, compute $q_t := \left(2 - \sum_{i=1}^{t-1} z_{\sigma(i)}\right)^{-1}$. Observe that $1/2 \leq q_t \leq 1$, as $0 \leq \sum_{i=1}^{k} z_i \leq 1$, and so the following OCRS is well-defined:

Algorithm 3 OCRS – Ezra et al. [23].

Require: $\boldsymbol{z} = (z_1, \ldots, z_k) \in \mathcal{P}$.
Ensure: at most one element of $[k]$.
for $t = 1, \ldots, k$ **do**
  if $\sigma(t) \in R(\boldsymbol{z})$ **then**
    Compute q_t based on the arrivals $\sigma(1), \ldots, \sigma(t-1)$.
    return $\sigma(t)$ independently with probability q_t.
  end if
end for
return $\emptyset$. ▷ pass on returning an element of $[k]$

▶ **Theorem 18** (Ezra et al. [34]). *Algorithm 3 is an OCRS which is* $1/2$*-selectable.*

Both Lee and Singla [34], as well as Adamczyk and Wlodarczyk [3], defined a special type of CRS called a **random order contention resolution scheme** (RCRS). Such a CRS is defined in the same way as an OCRS, except that the elements of $[k]$ arrive u.a.r. Suppose $Y_i \sim [0,1]$ u.a.r. and independently for $i = 1, \ldots, k$.

Algorithm 4 RCRS – Lee and Singla [34].

Require: $\boldsymbol{z} = (z_1, \ldots, z_k) \in \mathcal{P}$.
Ensure: at most one element of $[k]$.
for $i \in [k]$ in increasing order of Y_i **do**
  if $i \in R(\boldsymbol{z})$ **then**
    return i independently with probability $\exp(-Y_i \cdot z_i)$
  end if
end for
return $\emptyset$. ▷ pass on returning an element of $[k]$.

▶ **Theorem 19** (Lee and Singla [34]). *Algorithm 4 is a* $1 - 1/e$*-selectable RCRS.*

Suppose now $G = (U, V, E)$ is a known stochastic graph, whose online vertices $v_1, \ldots, v_n$ are presented according to the below algorithm via an adversarially chosen permutation $\pi : [n] \to [n]$ (i.e., $v_{\pi(1)}, \ldots, v_{\pi(n)}$). Let $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$ be an optimum solution to LP-config for G with edge variables $(\widetilde{x}_e)_{e \in E}$. For each $t \in [n]$ and $u \in U$, define $q_{u,t} := \left(2 - \sum_{i=1}^{t-1} z_{u, v_{\pi(i)}}\right)^{-1}$, where $z_e := p_e \widetilde{x}_e$ for $e \in E$, and $q_{u,1} := 1/2$. Clearly, $\sum_{v \in V} z_{u,v} \leq 1$, by constraint (1) of LP-config, and so $1/2 \leq q_{u,t} \leq 1$:

Algorithm 5 Known Stochastic Graph – AOM – Modified.

Require: a stochastic graph $G = (U, V, E)$.
Ensure: a matching $\mathcal{M}$ of G of active edges.
1: $\mathcal{M} \leftarrow \emptyset$.
2: Compute an optimum solution of LP-config for G, say $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$.
3: **for** $t = 1, \dots, n$ **do**
4: Based on the previous arrivals $v_{\pi(1)}, \dots, v_{\pi(t-1)}$ before $v_{\pi(t)}$, compute values $(q_{u,t})_{u \in U}$.
5: Set $e \leftarrow$ `VertexProbe` $\left(v_{\pi(t)}, \partial(v_{\pi(t)}), (x_{v_{\pi(t)}}(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_{v_{\pi(t)}}}\right)$.
6: **if** $e = (u, v_{\pi(t)})$ for some $u \in U$, and u is unmatched **then**
7: Add e to $\mathcal{M}$ independently with probability $q_{u,t}$. ▷ OCRS is used here
8: **end if**
9: **end for**
10: **return** $\mathcal{M}$.

▶ **Proposition 20.** *Algorithm 5 is* $1/2$*-competitive against adversarial arrivals.*

Proof. Given $u \in U$, let $\mathcal{M}(u)$ denote the edge matched to u by $\mathcal{M}$, where $\mathcal{M}(u) := \emptyset$ if no such edge exists. Observe now that if $C(e)$ corresponds to the event in which `VertexProbe` commits to $e \in \partial(u)$, then $\mathbb{P}[C(e)] = p_e \widetilde{x}_e$ by Lemma 10. Moreover, the events $(C(e))_{e \in \partial(u)}$ are independent, and satisfy

$$\sum_{e \in \partial(u)} \mathbb{P}[C(e)] = \sum_{e \in \partial(u)} p_e \widetilde{x}_e \leq 1, \tag{15}$$

by constraint (1) of LP-config. As such, denote $\boldsymbol{z} := (z_e)_{e \in \partial(u)}$ where $z_e = p_e \widetilde{x}_e$, and observe that (15) ensures that $\boldsymbol{z} \in \mathcal{P}$, where $\mathcal{P}$ is the convex relaxation of the rank one matroid on $\partial(u)$. Let us denote $R(\boldsymbol{z})$ as those those $e \in \partial(u)$ for which $C(e)$ occurs.

If ψ is the OCRS defined in Algorithm 3, then we may pass $\boldsymbol{z}$ to ψ, and process the edges of $\partial(u)$ in the order induced by π. Denote the resulting output by $\psi_{\boldsymbol{z}}(R(\boldsymbol{z}))$. By coupling the random draws of lines (4) and (7) of Algorithms 3 and 5, respectively, we get that

$$w(\mathcal{M}(u)) = \sum_{e \in \partial(u)} w_e \cdot \mathbf{1}_{[e \in R(\boldsymbol{z})]} \cdot \mathbf{1}_{[e \in \psi_{\boldsymbol{z}}(R(\boldsymbol{z}))]}$$

Thus, after taking expectations,

$$\mathbb{E}[w(\mathcal{M}(u))] = \sum_{e \in \partial(u)} w_e \cdot \mathbb{P}[e \in \psi_{\boldsymbol{z}}(R(\boldsymbol{z})) \,|\, e \in R(\boldsymbol{z})] \cdot \mathbb{P}[e \in R(\boldsymbol{z})].$$

Now, Theorem 18 ensures that for each $e \in \partial(u)$, $\mathbb{P}[e \in \psi_{\boldsymbol{z}}(R(\boldsymbol{z})) \,|\, e \in R(\boldsymbol{z})] \geq 1/2$. It follows that $\mathbb{E}[w(\mathcal{M}(u))] \geq \frac{1}{2} \sum_{e \in \partial(u)} w_e p_e \widetilde{x}_e$, for each $u \in U$. Thus,

$$\mathbb{E}[w(\mathcal{M})] = \sum_{u \in U} \mathbb{E}[w(\mathcal{M}(u))] \geq \frac{1}{2} \sum_{e \in E} w_e p_e \widetilde{x}_e = \frac{\text{LPOPT}(G)}{2},$$

where the equality follows since $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$ is an optimum solution to LP-config. On the other hand, $\text{LPOPT}(G) \geq \text{OPT}(G)$ by Theorem 8, and so the proof is complete. ◀

For each $v \in V$, draw $\widetilde{Y}_v \in [0, 1]$ independently and u.a.r. We assume that the vertices of V are presented to the below online probing algorithm in non-decreasing order according to the values $(\widetilde{Y}_v)_{v \in V}$. Note that this is equivalent to presenting V to the algorithm in random order.

Algorithm 6 Known Stochastic Graph – ROM– Modified.

Require: a stochastic graph $G = (U, V, E)$.
Ensure: a matching $\mathcal{M}$ of G of active edges.

```
M ← ∅.
Compute an optimum solution of LP-config for G, say (x_v(e))_{v∈V, e∈C_v}.
for s ∈ V in increasing order of Ỹ_s do
    Set e ← VertexProbe(s, ∂(s), (x_s(e))_{e∈C_s}).
    if e = (u, s) for some u ∈ U, and u is unmatched then
        Add e to M independently with probability exp(−Ỹ_s · p_{u,s} · x̃_{u,s}).
    end if
end for
return M.
```

▶ **Proposition 21** (Restatement of Corollary 5 and Remark 6)**.** *Algorithm 6 is non-adaptive and* $1 - 1/e$*-competitive against random order arrivals.*

Algorithm 6 is clearly non-adaptive, and the proof that it is $1 - 1/e$-competitive follows similarly to the proof of Proposition 20 (see [9] for the details).

5 Open problems

There are some basic questions that are unresolved. Perhaps the most basic question which is also unresolved in the classical setting without probing is to bridge the gap between the positive $1 - 1/e$ competitive ratio and in-approximations in the context of known i.d. random order arrivals. In terms of the single item prophet secretary problem (without probing), Correa et al. [20] obtain a 0.669 competitive ratio following Azar et al. [5] who were the first to surpass the $1 - 1/e$ “barrier”. Correa et al. [20] also establish a 0.732 in-approximation for the i.d. setting, and Huang et al. [32] recently established a 0.703 in-approximation for i.i.d. arrivals in the multi-item case. Can we surpass $1 - 1/e$ in the probing setting for i.d. input arrivals or for the special case of i.i.d. input arrivals? Is there a provable difference between stochastic bipartite matching (with probing constraints) and the classical online settings? Can we obtain the same competitive results against an optimal offline *non-committal* benchmark which respects the probing constraints but doesn’t operate in the probe-commit model? The 0.51 in-approximation result of Fata et al. [24] suggests that 0.51 may be the optimal competitive ratio against this stronger benchmark.

One interesting extension of the probing model is to allow non-Bernoulli edge random variables to describe edge uncertainty. Even for a single online vertex in the unconstrained setting, this problem is interesting as it corresponds to computing an optimal policy for the **free-order prophets problem**, which was recently studied by Segev and Singla in [37].

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, 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.

3 Marek Adamczyk and Michał Włodarczyk. Random order contention resolution schemes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 790–801. IEEE, 2018.

4 Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 18–35, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2229012.2229018.

5 Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Prophet secretary: Surpassing the 1-1/e barrier. In Éva Tardos, Edith Elkind, and Rakesh Vohra, editors, *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 303–318. ACM, 2018.

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. doi:10.1007/s00453-011-9511-8.

7 Alok Baveja, Amit Chavan, Andrei Nikiforov, Aravind Srinivasan, and Pan Xu. Improved bounds in stochastic matching and optimization. *Algorithmica*, 80(11):3225–3252, November 2018. doi:10.1007/s00453-017-0383-4.

8 Allan Borodin, Calum MacRury, and Akash Rakheja. Bipartite stochastic matching: Online, random order, and I.I.D. models. *CoRR*, abs/2004.14304, 2020. arXiv:2004.14304.

9 Allan Borodin, Calum MacRury, and Akash Rakheja. Prophet matching meets probing with commitment. *CoRR*, abs/2102.04325, 2021. arXiv:2102.04325.

10 Allan Borodin, Calum MacRury, and Akash Rakheja. Secretary matching meets probing with commitment. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 13:1–13:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

11 Brian Brubach, Nathaniel Grammel, Will Ma, and Aravind Srinivasan. Follow your star: New frameworks for online stochastic matching with known and unknown patience. *CoRR*, abs/1907.03963, 2021. arXiv:1907.03963.

12 Brian Brubach, Nathaniel Grammel, Will Ma, and Aravind Srinivasan. Follow your star: New frameworks for online stochastic matching with known and unknown patience. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2872–2880. PMLR, 13–15 April 2021. URL: http://proceedings.mlr.press/v130/brubach21a.html.

13 Brian Brubach, Nathaniel Grammel, Will Ma, and Aravind Srinivasan. Improved guarantees for offline stochastic matching via new ordered contention resolution schemes. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27184–27195, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/e43739bba7cdb577e9e3e4e42447f5a5-Abstract.html.

14 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.

15 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. doi:10.1007/s00453-019-00603-7.

16 Constantine Caramanis, Paul Dütting, Matthew Faw, Federico Fusco, Philip Lazos, Stefano Leonardi, Orestis Papadigenopoulos, Emmanouil Pountourakis, and Rebecca Reiffenhäuser. Single-sample prophet inequalities via greedy-ordered selection. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2022)*, pages 1298–1325, 2022. doi:10.1137/1.9781611977073.54.

17 Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 311–320. ACM, 2010.

18 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.

19 José R. Correa, Patricio Foncea, Dana Pizarro, and Victor Verdugo. From pricing to prophets, and back! *Oper. Res. Lett.*, 47(1):25–29, 2019.

20 José R. Correa, Raimundo Saona, and Bruno Ziliotto. Prophet secretary through blind strategies. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1946–1961, 2019.

21 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.

22 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, pages 700–714, USA, 2018. Society for Industrial and Applied Mathematics.

23 Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. Online stochastic max-weight matching: Prophet inequality for vertex and edge arrival models. In *Proceedings of the 21st ACM Conference on Economics and Computation*, EC '20, pages 769–787, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3391403.3399513.

24 Elaheh Fata, Will Ma, and David Simchi-Levi. Multi-stage and multi-customer assortment optimization with inventory constraints. *CoRR*, abs/1908.09808, 2019. arXiv:1908.09808.

25 Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1014–1033. SIAM, 2016. doi:10.1137/1.9781611974331.ch72.

26 Hu Fu, Zhihao Gavin Tang, Hongxun Wu, Jinzhao Wu, and Qianfan Zhang. Random Order Vertex Arrival Contention Resolution Schemes for Matching, with Applications. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 68:1–68:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2021.68.

27 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, pages 2841–2854, USA, 2019. Society for Industrial and Applied Mathematics.

28 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. doi:10.1145/1147954.1147956.

29 Bernd Gärtner and Jirí Matousek. *Understanding and using linear programming.* Universitext. Springer, 2007.

30 Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Algorithms and adaptivity gaps for stochastic probing. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1731–1747. SIAM, 2016. `doi:10.1137/1.9781611974331.ch120`.

31 Mohammad Taghi Hajiaghayi, Robert D. Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 58–65. AAAI Press, 2007.

32 Zhiyi Huang, Xinkai Shu, and Shuyi Yan. The power of multiple choices in online stochastic matching, 2022. `doi:10.48550/ARXIV.2203.02883`.

33 Haim Kaplan and David Naori dand Danny Raz. Online weighted bipartite matching with a sample. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2022)*, 2022. `doi:10.1137/1.9781611977073.52`.

34 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. `doi:10.4230/LIPIcs.ESA.2018.57`.

35 Tristan Pollner, Mohammad Roghani, Amin Saberi, and David Wajc. Improved online contention resolution for matchings and applications to the gig economy. In David M. Pennock, Ilya Segal, and Sven Seuken, editors, *EC '22: The 23rd ACM Conference on Economics and Computation, Boulder, CO, USA, July 11 - 15, 2022*, pages 321–322. ACM, 2022. `doi:10.1145/3490486.3538295`.

36 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. `doi:10.1002/bimj.4710320805`.

37 Danny Segev and Sahil Singla. Efficient approximation schemes for stochastic probing and prophet problems. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, pages 793–794, New York, NY, USA, 2021. Association for Computing Machinery. `doi:10.1145/3465456.3467614`.

38 Jan Vondrák, Chandra Chekuri, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 783–792, New York, NY, USA, 2011. Association for Computing Machinery. `doi:10.1145/1993636.1993740`.

A Extending to Known I.D. Arrivals

Suppose that $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$ is a known i.d. input, where $H_{\text{typ}} = (U, B, F)$ has downward-closed online probing constraints $(\mathcal{C}_b)_{b \in B}$. If $G \sim (H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$, where $G = (U, V, E)$ has vertices $V = \{v_1, \ldots, v_n\}$, then define $r_i(b) := \mathbb{P}[v_i = b]$ for each $i \in [n]$ and $b \in B$, where we hereby assume that $r_i(b) > 0$. We generalize LP-config to account for the distributions $(\mathcal{D}_i)_{i=1}^n$. For each $i \in [n], b \in B$ and $\boldsymbol{e} \in \mathcal{C}_b$, we introduce a decision variable $x_i(\boldsymbol{e} \,||\, b)$ to encode the probability that v_i has type b *and* $\boldsymbol{e}$ is the sequence of edges of $\partial(v_i)$ probed by the *relaxed* benchmark.

$$
\begin{aligned}
&\text{maximize} && \sum_{i\in[n],b\in B,\boldsymbol{e}\in\mathcal{C}_b} \mathrm{val}(\boldsymbol{e})\cdot x_i(\boldsymbol{e}\,||\,b) && \text{(LP-config-id)}\\
&\text{subject to} && \sum_{i\in[n],b\in B}\ \sum_{\substack{\boldsymbol{e}\in\mathcal{C}_b:\\(u,b)\in\boldsymbol{e}}} p_{u,b}\cdot q(\boldsymbol{e}_{<(u,b)})\cdot x_i(\boldsymbol{e}\,||\,b)\le 1 && \forall u\in U \quad (16)\\
& && \sum_{\boldsymbol{e}\in\mathcal{C}_b} x_i(\boldsymbol{e}\,||\,b) = r_i(b) && \forall b\in B, i\in[n] \quad (17)\\
& && x_i(\boldsymbol{e}\,||\,b)\ge 0 && \forall b\in B,\boldsymbol{e}\in\mathcal{C}_b, i\in[n] \quad (18)
\end{aligned}
$$

Let us denote $\mathrm{LPOPT}(H_{\mathrm{typ}}, (\mathcal{D}_i)_{i=1}^n)$ as the value of an optimum solution to LP-config-id.

▶ **Theorem 22.** *$OPT(H_{typ}, (\mathcal{D}_i)_{i=1}^n) \leq LPOPT(H_{typ}, (\mathcal{D}_i)_{i=1}^n)$.*

One way to prove Theorem 22 is to use the properties of the relaxed benchmark on G guaranteed by Lemma 14, and the above interpretation of the decision variables to argue that $\mathbb{E}[\mathrm{OPT}_{\mathrm{rel}}(G)] \leq \mathrm{LPOPT}(H_{\mathrm{typ}}, (\mathcal{D}_i)_{i=1}^n)$, where $\mathrm{OPT}_{\mathrm{rel}}(G)$ is the value of the relaxed benchmark on G. Specifically, we can interpret (16) as saying that the relaxed benchmark matches each offline vertex at most once in expectation. Moreover, (17) holds by observing that if v_i is of type b, then the relaxed benchmark selects some $\boldsymbol{e} \in \mathcal{C}_b$ to probe (note $\boldsymbol{e}$ could be the empty-string). We provide a morally equivalent proof of Theorem 22 in [9]. Specifically, we consider an optimum solution of LP-config with respect to G, and apply a conditioning argument in conjunction with Theorem 8.

Given a feasible solution to LP-config-id, say $(x_i(\boldsymbol{e}\,||\,b))_{i\in[n],b\in B,\boldsymbol{e}\in\mathcal{C}_b}$, for each $u \in U, i \in [n]$ and $b \in B$ define

$$\widetilde{x}_{u,i}(b) := \sum_{\substack{\boldsymbol{e}\in\mathcal{C}_b:\\(u,b)\in\boldsymbol{e}}} q(\boldsymbol{e}_{<(u,b)})\cdot x_i(\boldsymbol{e}\,||\,b). \tag{19}$$

We refer to $\widetilde{x}_{u,i}(b)$ as an **edge variable**, thus extending the definition from the known stochastic graph setting. Suppose now that we fix $i \in [n]$ and $b \in B$, and consider the variables, $(x_i(\boldsymbol{e}\,||\,b))_{\boldsymbol{e}\in\mathcal{C}_b}$. Observe that (17) ensures that $\frac{\sum_{\boldsymbol{e}\in\mathcal{C}_b} x_i(\boldsymbol{e}\,||\,b)}{r_i(b)} = 1$. Hence, regardless of which type node v_i is drawn as, $\frac{\sum_{\boldsymbol{e}\in\mathcal{C}_{v_i}} x_i(\boldsymbol{e}\,||\,v_i)}{r_i(v_i)} = 1$. We can therefore generalize `VertexProbe` as follows. Given vertex v_i, draw $\boldsymbol{e}' \in \mathcal{C}_{v_i}$ with probability $x_i(\boldsymbol{e}'\,||\,v_i)/r_i(v_i)$. If $\boldsymbol{e}' = \lambda$, then return the empty-set. Otherwise, set $\boldsymbol{e}' = (e'_1, \ldots, e'_k)$ for $k := |\boldsymbol{e}'| \geq 1$, and probe the edges of $\boldsymbol{e}'$ in order. Return the first edge which is revealed to be active, if such an edge exists. Otherwise, return the empty-set. We denote the output of `VertexProbe` on the input $(v_i, \partial(v_i), (x_i(\boldsymbol{e}\,||\,v_i)/r_i(v_i))_{\boldsymbol{e}\in\mathcal{C}_{v_i}})$ by `VertexProbe`$(v_i, \partial(v_i), (x_i(\boldsymbol{e}\,||\,v_i)/r_i(v_i))_{\boldsymbol{e}\in\mathcal{C}_{v_i}})$. Define $C(u, v_i)$ as the event in which `VertexProbe` outputs the edge (u, v_i), and observe the following extension of Lemma 10:

▶ **Lemma 23.** *If* `VertexProbe` *is passed* $\big(v_i, \partial(v_i), (x_i(\boldsymbol{e}\,||\,v_i)/r_i(v_i))_{\boldsymbol{e}\in\mathcal{C}_{v_i}}\big)$*, then for any $b \in B$ and $u \in U$, $\mathbb{P}[C(u, v_i) \,|\, v_i = b] = \frac{p_{u,b}\cdot\widetilde{x}_{u,i}(b)}{r_i(b)}$.*

▶ Remark 24. As in Definition 12, if $C(u, v_i)$ occurs, then u **commits** to (u, v_i) (or v_i).

We now generalize Algorithm 1 where π is generated either u.a.r. or adversarially.

Algorithm 7 Known I.D.

```
Require: a known i.d. input (H_typ, (D_i)_{i=1}^n).
Ensure: a matching M of active edges of G ~ (H_typ, (D_i)_{i=1}^n).
M ← ∅.
Compute an optimum solution of LP-config-id for (H_typ, (D_i)_{i=1}^n), say
  (x_i(e || b))_{i∈[n], b∈B, e∈C_b}.
for t = 1, ..., n do
    Let a ∈ B be the type of the current arrival v_{π(t)}.        ▷ to simplify notation
    Set e ← VertexProbe(v_{π(t)}, ∂(v_{π(t)}), (x_{π(t)}(e || a) · r_{π(t)}^{-1}(a))_{e∈C_a}).
    if e = (u, v_{π(t)}) for some u ∈ U, and u is unmatched then
        Add e to M.
    end if
end for
return M.
```

Similarly, to Algorithm 1, one can show that Algorithm 7 attains a competitive ratio of 1/2 for random order arrivals. Interestingly, if the distributions $(\mathcal{D}_i)_{i=1}^n$ are identical – that is, we work with known i.i.d. arrivals – then it is relatively easy to show that this algorithm becomes $1-1/e$-competitive.

▶ **Proposition 25.** *If Algorithm 7 is presented a known i.i.d. input, say the type graph H_{typ} together with the distribution $\mathcal{D}$, then $\mathbb{E}[w(\mathcal{M})] \geq (1-1/e)\, OPT(H_{typ}, \mathcal{D})$.*

▶ Remark 26. Proposition 25 is proven explicitly in an earlier 2020 arXiv version of this paper for the case of patience values.

Returning to the case of non-identical distributions, observe that in the execution of Algorithm 7 the probability that v_i commits to the edge (u, v_i) for $u \in U$ is precisely

$$z_{u,i} := \sum_{b \in B} p_{u,b} \cdot \widetilde{x}_{u,i}(b) = \sum_{b \in B} \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_b: \\ (u,b) \in \boldsymbol{e}}} p_{u,b} \cdot q(\boldsymbol{e}_{<(u,b)}) \cdot x_i(\boldsymbol{e} \,||\, b). \tag{20}$$

Moreover, the events $(C(u, v_i))_{i=1}^n$ are independent, so this suggests applying the same contention resolutions schemes as in the known stochastic graph setting. We first focus on the adversarial arrival model, where we assume the vertices $v_1, \ldots, v_n$ are presented in some unknown order $\pi : [n] \to [n]$. We make use of the OCRS from before (Algorithm 3). For each $t \in [n]$ and $u \in U$, define

$$q_{u,t} := \frac{1}{2 - \sum_{i=1}^{t-1} z_{u,\pi(i)}}, \tag{21}$$

where $q_{u,1} := 1/2$. Note that $1/2 \leq q_{u,t} \leq 1$ as $\sum_{j \in [n]} z_{u,j} \leq 1$ by constraint (16) of LP-config-id. We define Algorithm 8 by modifying Algorithm 7 using the OCRS to ensure that each $i \in [n]$ is matched to $u \in U$ with probability $z_{u,i}/2$. However, to achieve a competitive ratio of 1/2, we require the stronger claim that for each type node $a \in B$, the probability (u, v_i) is added to the matching *and* v_i is of type a is lower bounded by $p_{u,a}\widetilde{x}_{u,i}(a)/2$. Crucially, if we condition on $u \in U$ being unmatched when v_i is processed, v_i having type a, and $C(u, v_i)$, then the probability the OCRS matches u to v_i does *not* depend on a. This implies the desired lower bound of $p_{u,a}\widetilde{x}_{u,i}(a)/2$, and so Algorithm 8 attains a competitive ratio of 1/2 by (19) and Theorem 22 (we provide the details in the proof below).

Algorithm 8 Known I.D. – AOM – Modified.

Require: a known i.d. input $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$.
Ensure: a matching $\mathcal{M}$ of active edges of $G \sim (H_{\text{typ}}, (\mathcal{D}_t)_{t=1}^n)$.
$\mathcal{M} \leftarrow \emptyset$.
Compute an optimum solution of LP-config-id for $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$, say $(x_i(\boldsymbol{e} \,||\, b))_{i \in [n], b \in B, \boldsymbol{e} \in \mathcal{C}_b}$.
for $t = 1, \ldots, n$ **do**
  Let $a \in B$ be the type of the current arrival $v_{\pi(t)}$.
  Based on the previous arrivals $v_{\pi(1)}, \ldots, v_{\pi(t-1)}$ before $v_{\pi(t)}$, compute values $(q_{u,t})_{u \in U}$.
  Set $e \leftarrow \texttt{VertexProbe}\left(v_{\pi(t)}, \partial(v_{\pi(t)}), \left(x_{\pi(t)}(\boldsymbol{e} \,||\, a) \cdot r^{-1}_{\pi(t)}(a)\right)_{\boldsymbol{e} \in \mathcal{C}_a}\right)$.
  if $e = (u, v_t)$ for some $u \in U$, and u is unmatched **then**
    Add e to $\mathcal{M}$ independently with probability $q_{u,t}$.
  end if
end for
return $\mathcal{M}$.

Proof of Theorem 1. For notational simplicity, let us assume that $\pi(t) = t$ for each $t \in [n]$, so that the online vertices arrive in order $v_1, \ldots, v_n$. Now, the edge variables $(\widetilde{x}_{u,t}(b))_{u \in U, t \in [n], b \in B}$ satisfy $\text{LPOPT}(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n) = \sum_{u \in U, t \in [n], b \in B} p_{u,b} w_{u,b} \widetilde{x}_{u,t}(b)$. Thus, to complete the proof it suffices to show that

$$\mathbb{P}[(u, v_t) \in \mathcal{M} \text{ and } v_t = b] \geq \frac{\widetilde{x}_{u,t}(b)}{2} \tag{22}$$

for each $u \in U, t \in [n]$ and $b \in B$, where we hereby assume w.l.o.g. that $\widetilde{x}_{u,t}(b) > 0$. In order to prove this, we first observe that by the same coupling argument used in the proof of Proposition 20,

$$\mathbb{P}[(u, v_t) \in \mathcal{M}] \geq \frac{z_{u,t}}{2} = \frac{1}{2} \sum_{b \in B} p_{u,b} \widetilde{x}_{u,t}(b) \tag{23}$$

as a result of the 1/2-selectability of Algorithm 3. Let us now define R_t as the unmatched vertices of U when v_t arrives. Observe then that

$$\mathbb{P}[(u, v_t) \in \mathcal{M} \mid v_t = b, C(u, v_t) \text{ and } u \in R_t] = q_{u,t}. \tag{24}$$

Now, $\mathbb{P}[v_t = b, C(u, v_t) \text{ and } u \in R_t] = p_{u,b} \cdot \widetilde{x}_{u,t}(b) \cdot \mathbb{P}[u \in R_t]$, by Lemma 23 and the independence of the events $\{v_t = b\} \cap \{C(u, v_t)\}$ and $\{u \in R_t\}$. Thus, by the law of total probability,

$$\sum_{b \in B} p_{u,b} \widetilde{x}_{u,t} q_{u,t} \cdot \mathbb{P}[u \in R_t] = \mathbb{P}[(u, v_t) \in \mathcal{M}] \geq \frac{z_{u,t}}{2} = \frac{1}{2} \sum_{b \in B} p_{u,b} \widetilde{x}_{u,t}(b)$$

where the second inequality follows from (23). Thus, $q_{u,t} \cdot \mathbb{P}[u \in R_t] \geq 1/2$, and so combined with (24), (22) follows, thus completing the proof. ◀

Suppose now that each vertex v_t has an arrival time, say $\widetilde{Y}_t \in [0, 1]$, drawn u.a.r. and independently for $t \in [n]$. The values $(\widetilde{Y}_t)_{t=1}^n$ indicate the increasing order in which the vertices $v_1, \ldots, v_n$ arrive.

Algorithm 9 Known I.D. – ROM – Modified.

Require: a known i.d. input $(H_{\text{typ}}, (\mathcal{D}_t)_{t=1}^n)$.
Ensure: a matching $\mathcal{M}$ of active edges of $G \sim (H_{\text{typ}}, (\mathcal{D}_t)_{t=1}^n)$.

$\mathcal{M} \leftarrow \emptyset$.
Compute an optimum solution of LP-config-id for $(H_{\text{typ}}, (\mathcal{D}_t)_{t=1}^n)$, say $(x_t(\boldsymbol{e} \,||\, b))_{t \in [n], b \in B, \boldsymbol{e} \in \mathcal{C}_b}$.
for $t \in [n]$ in increasing order of $\widetilde{Y}_t$ **do**
  Set $e \leftarrow \texttt{VertexProbe}\left(v_t, \partial(v_t), (x_t(\boldsymbol{e} \,||\, v_t)/r_t(v_t))_{\boldsymbol{e} \in \mathcal{C}_{v_t}}\right)$.
  if $e = (u, v_t)$ for some $u \in U$, and u is unmatched **then**
    Add e to $\mathcal{M}$ independently with probability $\exp(-\widetilde{Y}_t \cdot z_{u,t})$.
  end if
end for
return $\mathcal{M}$.

Proof of Theorem 3. The competitive ratio of $1 - 1/e$ follows by the same coupling argument as in Proposition 21, together with the same observations used in the proof of Theorem 1, and so we omit the argument. ◀

B LP Relations

Suppose that we are given an arbitrary stochastic graph $G = (U, V, E)$. In this section, we first prove the equivalence between the relaxed stochastic matching problem and LP-config. We then state LP-std, the standard LP in the stochastic matching literature, as introduced by Bansal et al. [6], as well as LP-QC, the LP introduced by Gamlath et al. [27]. We then show that LP-QC and LP-config have the same optimum value when G is unconstrained.

▶ **Theorem 27.** $\mathit{OPT}(G) = \mathit{LPOPT}_{\mathit{conf}}(G)$

Proof. Clearly, Theorem 8 accounts for one side of the inequality, so it suffices to show that $\text{LPOPT}(G) \leq \text{OPT}_{\text{rel}}(G)$. Suppose we are presented a feasible solution $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$ to LP-config. Consider then the following algorithm:

1. $\mathcal{M} \leftarrow \emptyset$.
2. For each $v \in V$, set $e \leftarrow \texttt{VertexProbe}(v, \partial(v), (x_v(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_v})$. If $e \neq \emptyset$, then add e to $\mathcal{M}$.
3. Return $\mathcal{M}$.

Using Lemma 10, it is clear that $\mathbb{E}[w(\mathcal{M})] = \sum_{v \in V} \sum_{\boldsymbol{e} \in \mathcal{C}_v} \text{val}(\boldsymbol{e}) \cdot x_v(\boldsymbol{e})$. Moreover, each vertex $u \in U$ is matched by $\mathcal{M}$ at most once in expectation, as a consequence of constraint (1) of LP-config, and so the algorithm satisfies the required properties of a relaxed probing algorithm. The proof is therefore complete. ◀

Consider LP-std, which is defined only when G has patience values $(\ell_v)_{v \in V}$. Here each $e \in E$ has a variable x_e corresponding to the probability that the offline adaptive benchmark probes e.

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

$$\text{subject to} \quad \sum_{e \partial(u)} p_e \cdot x_e \leq 1 \qquad \forall u \in U \tag{25}$$

$$\sum_{e \in \partial(v)} p_e \cdot x_e \leq 1 \qquad \forall v \in V \tag{26}$$

$$\sum_{e \in \partial(v)} x_e \leq \ell_v \qquad \forall v \in V \tag{27}$$

$$0 \leq x_e \leq 1 \qquad \forall e \in E. \tag{28}$$

Gamlath et al. modified LP-std in the unconstrained setting by adding in exponentially many extra constraints. Specifically, for each $v \in V$ and $S \subseteq \partial(v)$, they ensure that

$$\sum_{e \in S} p_e \cdot x_e \leq 1 - \prod_{e \in S}(1 - p_e), \tag{29}$$

In the same variable interpretation as LP-std, the left-hand side of (29) corresponds to the probability the adaptive benchmark matches an edge of $S \subseteq \partial(v)$, and the right-hand side corresponds to the probability an edge of S is active[2].

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

$$\text{subject to} \quad \sum_{e \in S} p_e \cdot x_e \leq 1 - \prod_{e \in S}(1 - p_e) \qquad \forall v \in V, S \subseteq \partial(v) \tag{30}$$

$$\sum_{e \in \partial(u)} p_e \cdot x_e \leq 1 \qquad \forall u \in U \tag{31}$$

$$x_e \geq 0 \qquad \forall e \in E. \tag{32}$$

Let us denote $\text{LPOPT}_{\text{QC}}(G)$ as the optimum value of LP-QC.

▶ **Proposition 28.** *If G is unconstrained, then $LPOPT_{QC}(G) = LPOPT(G)$.*

In order to prove Proposition 28, we make use of a result of Gamlath et al. We mention that an almost identical result is also proven by Costello et al. [21] using different techniques.

▶ **Theorem 29** ([27])**.** *Suppose that $G = (U, V, E)$ is an unconstrained stochastic graph, and $(x_e)_{e \in E}$ is a solution to LP-QC. For each $v \in V$, there exists an online probing algorithm $\mathcal{B}_v$ whose input is $(v, \partial(v), (x_e)_{e \in \partial(v)})$, and which satisfies $\mathbb{P}[\mathcal{B}_v$ matches v to $e] = p_e x_e$ for each $e \in \partial(v)$.*

Proof of Proposition 28. Observe that by Theorem 27, in order to prove the claim it suffices to show that $\text{LPOPT}_{\text{QC}}(G) = \text{OPT}_{\text{rel}}(G)$. Clearly, $\text{OPT}_{\text{rel}}(G) \leq \text{LPOPT}_{\text{QC}}(G)$, as can be seen by defining x_e as the probability that the relaxed benchmark probes the edge $e \in E$. Thus, we focus on showing that $\text{LPOPT}_{\text{QC}}(G) \leq \text{OPT}_{\text{rel}}(G)$. Suppose that $(x_e)_{e \in E}$ is an optimum solution to $\text{LPOPT}_{\text{QC}}(G)$. We design the following algorithm, which we denote by $\mathcal{B}$:

1. $\mathcal{M} \leftarrow \emptyset$.
2. For each $v \in V$, execute $\mathcal{B}_v$ on $(v, \partial(v), (x_e)_{e \in \partial(v)})$, where $\mathcal{B}_v$ is the online probing algorithm of Theorem 29. If $\mathcal{B}_v$ matches v, then let e' be this edge, and add e' to $\mathcal{M}$
3. Return $\mathcal{M}$.

Using Theorem 29, it is clear that $\mathbb{E}[w(\mathcal{M})] = \sum_{e \in E} w_e p_e x_e$. Moreover, each vertex $u \in U$ is matched by $\mathcal{M}$ at most once in expectation, as a consequence of constraint (32). As a result, $\mathcal{B}$ is a relaxed probing algorithm. Thus, $\text{LPOPT}_{\text{QC}}(G) = \sum_{e \in E} w_e p_e x_e \leq \text{OPT}_{\text{rel}}(G)$, and so the proof is complete. ◀

[2] The LP considered by Gamlath et al. in [27] also places the analogous constraints of (29) on the vertices of U. That being said, these additional constraints are not used anywhere in the work of Gamlath et al., so we omit them.