# Online Bipartite Matching in the Probe-Commit Model[*]

Allan Borodin [†] Calum MacRury [‡]

**Abstract**

We consider the classical online bipartite matching problem in the probe-commit model. In this problem, when an online vertex arrives, its edges must be probed to determine if they exist, based on known edge probabilities. A probing algorithm must respect commitment, meaning that if a probed edge exists, it must be used in the matching. Additionally, each online vertex has a patience constraint which limits the number of probes that can be made to its adjacent edges. We introduce a new configuration linear program and use it to establish the following competitive ratios which depend on the model used to generate the instance graph, and the arrival order of its online vertices:

- In the worst-case instance model, an optimal $1/e$ ratio when the vertices arrive in uniformly at random (u.a.r.) order.
- In the known independently distributed (i.d.) instance model, an optimal $1/2$ ratio when the vertices arrive in adversarial order, and a $1-1/e$ ratio when the vertices arrive in u.a.r. order.

The latter two results improve upon the previous best competitive ratio of 0.46 due to Brubach et al. (Algorithmica 2020). Our $1 - 1/e$-competitive algorithm matches the best known result for the prophet secretary matching problem due to Ehsani et al. (SODA 2018). Our algorithm is efficient and implies a $1-1/e$ approximation ratio for the special case when the graph is known. This is the offline stochastic matching problem, and we improve upon the 0.42 approximation ratio for one-sided patience due to Pollner et al. (EC 2022), while also generalizing the $1 - 1/e$ approximation ratio for unbounded patience due to Gamlath et al. (SODA 2019).

[*]Journal paper is based on the following two conference papers: "Prophet Matching in the Probe-Commit Model" [9] and "Secretary Matching Meets Probing with Commitment" [10]

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

[‡]Graduate School of Business, Columbia University, New York. Email Address: `cm4379@columbia.edu`

## Contents

**1 Introduction** **3**

**2 Preliminaries** **4**
- 2.1 Benchmark . . . 6

**3 Contributions and Techniques** **6**
- 3.1 Generalizing to Downward Closed Online Probing Constraints . . . 9
- 3.2 An Overview of Our Techniques . . . 10
  - 3.2.1 Extending to Known I.D. Instances: . . . 14
  - 3.2.2 Worst-Case Instances and Random-Order Arrivals. . . . 14
  - 3.2.3 Solving LP-config Efficiently and Vertex-weighted Worst-case Instances . . . 14

**4 Relaxing the Offline Adaptive Benchmark via LP-config** **15**
- 4.1 Proving Lemma 4.1 . . . 16

**5 Known I.D. Instance Model** **20**

**6 Edge-weighted Worst-case Instance Model** **24**

**7 Vertex-weighted Worst-case Instance Model** **26**
- 7.1 Analyzing Algorithm 7 in the ROM . . . 29
- 7.2 Proving Theorem 7.7 . . . 31
  - 7.2.1 Proving Dual Feasibility: Lemma 7.12 . . . 32

**8 Efficiency of Our Algorithms** **35**

**9 A Tight Adaptivity Gap** **36**

**10 Conclusion and Open Problems** **38**

**A LP Relations** **43**

**B Deferred Proofs** **45**

## 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 which 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.

While we are only considering "one-sided online bipartite matching", offline stochastic matching was first considered in the context of a general graph by Chen et al. [17]. In this problem, *the probing algorithm* is presented a *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 $\ell_v$ associated with each vertex $v$. An algorithm *probes* edges in $E$ in some adaptive order within the constraint that at most $\ell_v$ edges are probed incident to any particular vertex $v$. 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. Chen et al. showed that by greedily probing edges in non-increasing order of edge probability, one attains an approximation ratio of 1/4 against an *optimal offline probing algorithm* (we provide a precise definition in Subsection 2.1). The analysis was later improved by Adamczyk [1], who showed that this greedy algorithm in fact attains an approximation ratio of 1/2.

In addition to generalizing the results of Chen et al. to edge weights, Bansal et al. [7] introduced a bipartite online variant of the (offline) stochastic matching problem called the *online stochastic matching problem with known i.i.d. arrivals*. 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 how many buyers will arrive, and the possible *type/profile* of each online buyer, which is specified by edge probabilities, edge weights and a patience parameter. Here the 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. Note that there are no restrictions on how many buyers an item may be shown to. 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).

Mehta and Panigrahi [41] introduced the *online matching problem with stochastic rewards*, which studies the original online bipartite matching problem of Karp, Vazirani and Vazirani [34] in the probe-commit model. Here the algorithm initially only knows the offline vertices $U$ of the stochastic graph $G = (U, V, E)$, and its online vertices $V$ are determined by an adversary and presented to the algorithm one by one. Moreover, they consider when $G$ has *unit patience values*, that is, each online vertex can probe at most one of its adjacent edges. They considered the special case of uniform edge probabilities (i.e, $p_e = p$ for all $e \in E$) and proved a competitive ratio $\frac{1}{2}(1 + (1 - p)^{2/p})$ (this limits to $\frac{1}{2}(1 + e^{-2}) \approx .567$ as $p \to 0$). Mehta et al. [42] considered arbitrary edge probabilities, and attained a competitive ratio of 0.534, and recently, Huang and Zhang [33] additionally handled the case of arbitrary offline vertex weights, while improving this ratio to 0.572. However, as in [41], both [42] and [33] require edge probabilities which are *vanishingly small*[1]. Goyal and Udwani [29] improved on both of these works by proving a 0.596 competitive ratio in the same setting. They also showed that $1 - 1/e$ is attainable for the separate special case of *vertex-decomposable*[2] edge probabilities. It remains open whether $1 - 1/e$ is attainable for arbitrary edge probabilities. Goyal and Udwani discuss the difficulty of this problem in the context of the Adwords problem with an arbitrary budget to bid ratio.

Clearly, any classical online matching problem can be generalized to the probe-commit model. Given such a problem, we can ask if the optimal competitive ratio is the same when probing is not required. When the optimal competitive ratio is not known (even in the classical setting), we can still ask whether there exists an online probing algorithm whose competitive ratio is equal to the best known competitive ratio when probing is not required. We provide a number of positive answers to questions of this form. In particular, we generalize the problem of Bansal et al. to *known i.d. arrivals*. 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. [5]. When online buyers arrive in random order, we generalize the *prophet secretary matching problem* of Ehsani et al. [22]. We prove a $1/2$ competitive ratio for adversarial arrivals and a $1 - 1/e$ competitive ratio for random order arrivals. These competitive ratios match the best known results when probing is not required (see [5, 22]), and the $1/2$ result is in fact tight. Note that the arrival order does not matter for the case of identical distributions, and so our $1 - 1/e$ result implies a $1 - 1/e$ competitive ratio for known i.i.d. arrivals. Up until very recently, this result also matched the best known competitive ratio when probing is not required due to Manshadi et al. [40]. Yan [49] improved on [40] and showed that a competitive ratio of $0.645 > 1 - 1/e$ is attainable for known i.i.d. arrivals when probing is not required. We are also the first to study the online stochastic matching problem with *worst-case random order arrivals*. Here the stochastic graph is adversarially generated, and its online vertices arrive in random order. When the graph is edge-weighted, we generalize the *secretary matching problem* [35] to the probe-commit model, and prove a competitive ratio of $1/e$, which is exactly the optimal competitive ratio when probing is not required.

All of the above competitive ratios in fact hold in a more general probing model, where for each $v \in V$ the patience value $\ell_v$ is generalized to a downward-closed *online probing constraint* $\mathcal{C}_v$, which specifies which sequences of edges adjacent to $v$ may be probed. For instance, this includes when $v$ has a *budget* and *edge probing costs* (i.e., $\mathcal{C}_v$ is a *knapsack constraint*). Probing constraints of a similar nature were originally considered by Gupta and Nagarajan [30] for a wide range of offline stochastic probing problems. For simplicity, we defer the precise statement of our probing model to Subsection 3.1, and first introduce the online stochastic matching problem restricted to patience values.

## 2 Preliminaries

An instance of the **online stochastic matching problem** is a **stochastic graph** specified in the following way. Let $G = (U, V, E)$ be a bipartite graph with edge weights $(w_e)_{e \in E}$ and edge probabilities $(p_e)_{e \in E}$, where $\partial(r)$ denotes the edges of $G$ which include $r$ for $r \in U \cup V$. Each $e \in E$

[1]Vanishingly small edge probabilities must satisfy $\max_{e \in E} p_e \to 0$, where the asymptotics are with respect to the size of $G$.

[2]Vertex-decomposable means that there exists probabilities $(p_u)_{u \in U}$ and $(p_v)_{v \in V}$, such that $p_{(u,v)} = p_u \cdot p_v$ for each $(u, v) \in E$.

is **active** independently with probability $p_e$, where the **edge state** $\text{st}(e) \sim \text{Ber}(p_e)$ is the indicator random variable for this event. In addition, each **online vertex** $v \in V$ has an integer **patience parameter** $\ell_v \geq 1$. We denote $n := |V|$ to be the number of online vertices of $G$.

An **online probing algorithm** begins with limited information regarding $G$. Specifically, it knows $U$, the **offline vertices** of $G$, and in all but one of the settings we study, it also knows the value of $n$. An ordering on $V$ is then generated either by an **adversary** or **uniformly at random**, independent of all other randomization. We refer to the former case as the **adversarial order model (AOM)** and the latter case as the **random order model (ROM)**.

In the **worst-case instance model**, the stochastic graph $G$ is generated by an adversary. Based on whichever ordering is generated on $V$, the online vertices are then presented to the online probing algorithm one by one. When an online node $v \in V$ is presented (arrives), the online probing algorithm learns $(p_e)_{e \in \partial(v)}$ and $(w_e)_{e \in \partial(v)}$, however, the edge states $(\text{st}(e))_{e \in \partial(v)}$ initially remain hidden to the algorithm. Instead, the algorithm also learns the patience value $\ell_v$ of $v$, and it is allowed to *adaptively* **probe** at most $\ell_v$ edges of $\partial(v)$. Here a probe to an edge $e \in \partial(v)$ reveals the *instantiation* of $\text{st}(e)$ to the algorithm. 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. [31], and is slightly different but equivalent to the Chen et al. [17] model in which an active edge must be immediately accepted into the matching. As in the classical bipartite matching problem, an online probing algorithm must decide on a possible match for an online node $v$ before seeing the next online node. The algorithm returns a matching $\mathcal{M}$ of (probed) active edges, and its goal is to maximize $\mathbb{E}[w(\mathcal{M})]$, where $w(\mathcal{M}) := \sum_{e \in \mathcal{M}} w_e$ is the **weight** of $\mathcal{M}$, and the expectation is over $(\text{st}(e))_{e \in E}$ and any random decisions of the algorithm (as well as the ordering on $V$ in the ROM). We will consider online algorithms which know the value of $n = |V|$, as well as those which do not. By setting edge probabilities to 0, we hereby assume $E = U \times V$ without loss in generality.

In the **known i.d. instance model**, the algorithm again executes on an unknown stochastic graph $G = (U, V, E)$, however the online vertices $V$ and edges $E$ of $G$ are instead generated through the following stochastic process: Let $H_{\text{typ}} = (U, B, F)$ be a **type graph**, which is a bipartite graph with edge weights $(w_f)_{f \in F}$, edge probabilities $(p_f)_{f \in F}$, and patience values $(\ell_b)_{b \in B}$. We refer to $B$ as the **type nodes** of $H_{\text{typ}}$, as $H_{\text{typ}}$ is known to the algorithm and these represent the possible online vertices that $G$ may have. In addition, the algorithm knows that $n = |V|$ online vertices will arrive, and it is presented distributions $(\mathcal{D}_i)_{i=1}^n$ supported on $B$. For $i = 1, \ldots, n$, online vertex $v_i$ is drawn independently from $\mathcal{D}_i$, and $V$ is now defined to be the multiset including $v_1, \ldots, v_n$. The online vertices $V$ are once again presented to the algorithm either in adversarial or random order, and processed in the same way as in the worst-case instance model, with the caveat that the online algorithm additionally learns the distribution the online vertex was drawn from upon its arrival. We denote $G \sim (H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$ to indicate that $G$ is drawn from the **known i.d. input** $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$. The algorithm returns a matching $\mathcal{M}$ of (probed) active edges of $G$, and its goal is to maximize $\mathbb{E}[w(\mathcal{M})]$, where the expectation is now also over the additional randomness in $G \sim (H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$. Note that in the AOM, we assume that the ordering is generated by an **oblivious adversary**. This means that the ordering is a permutation $\pi$ of $[n] = \{1, \ldots, n\}$ which depends solely on $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$. The vertices $v_1, \ldots, v_n$ then arrive in order $v_{\pi(1)}, \ldots, v_{\pi(n)}$. We again hereby assume that $F = U \times B$.

## 2.1 Benchmark

It is easy to see 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* [7, 3, 15, 16]. An **offline probing algorithm** is given the stochastic graph $G = (U, V, E)$, but initially the edge states $(\text{st}(e))_{e \in E}$ are hidden. Its goal is to construct a matching of (probed) active edges of $G$ with weight as large as possible in expectation. It can adaptively probe the edges of $E$ in any order, potentially interleaving edges between distinct vertices. For instance, it may probe $e_1 \in \partial(v)$ followed by $e_2 \in \partial(v')$ and then $e_3 \in \partial(v)$ for distinct $v, v' \in V$. However, at most $\ell_v$ edges of $\partial(v)$ may be probed for each $v \in V$, and it must operate in the same probe-commit model as an online probing algorithm. We define the **(offline) adaptive benchmark**, denoted OPT, to be an optimal offline probing algorithm, and define $\text{OPT}(G)$ to be the expected weight of the matching returned by OPT when it executes on $G$. An alternative weaker benchmark used by Brubach et al. [13, 11] 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 $e_2 = (u, v') \in E$ is probed after $e_1 = (u, v)$ for $v' \neq v$, then no edge of $\partial(v)$ is probed following $e_2$.

In this work, we focus exclusively on the offline adaptive benchmark. In the worst-case instance model, we benchmark against $\text{OPT}(G)$ for a worst-case stochastic graph $G$. In the known i.d. instance model, we benchmark against $\text{OPT}(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n) := \mathbb{E}[\text{OPT}(G)]$, where the expectation is over the randomness in drawing $G$ from a worst-case known i.d. input $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$. In either instance model, we can benchmark against a restricted sub-class of instances. This is relevant to us in the worst-case instance model, where in one case we benchmark against **(offline) vertex-weighted** stochastic graphs (i.e., there exists $(w_u)_{u \in U}$ such that $w_{u,v} = w_u$ for all $(u, v) \in E$).

Even assuming the full generality of edge-weighted stochastic graphs, we are still left with a wide range of problems, depending on if we work in the worst-case instance model or the known i.d. instance model, as well as whether we assume adversarial or random order arrivals. We refer to each problem as the **online stochastic matching problem** with a **worst-case (known i.d.) instance** and **adversarial (random order) arrivals**. If we restrict to a sub-class of stochastic graphs, then we indicate this in the middle of the problem name. For instance, when $G$ is restricted to vertex weights, we refer to the problem as the online stochastic matching problem with a worst-case *vertex-weighted* instance and random order arrivals.

## 3 Contributions and Techniques

We first summarize the competitive ratios of our algorithms in Table 1. We then discuss each result individually and explain its significance. Afterwards, in Subsection 3.1, we describe the general probing model from Section 1. All our algorithms are efficient in this model, as we prove in Section 8. With the exception of our vertex-weighted algorithm, all our algorithms can be extended to this model without a loss in competitiveness. In Subsection 3.2, we give an overview of the techniques used in the paper.

| Competitive Ratios | AOM | ROM |
|---|---|---|
| Known I.D. Instance | ? $\to\geq$ **1/2** [§5]<br>$\leq 1/2$ [36] | ? $\to$ **1-1/e** [§5]<br>$\leq 0.703$ [32] |
| Known I.I.D. Instance | $\geq 0.46$ [16] $\to\geq$ **1-1/e** [§5]<br>$\leq 0.703$ [32] | $\geq 0.46$ [16] $\to\geq$ **1-1/e** [§5]<br>$\leq 0.703$ [32] |
| Worst-case<br>Edge-weighted Instance | $-$ | ? $\to\geq$ **1/e** [§6]<br>$\leq 1/e$ [38] |
| Worst-case<br>Vertex-weighted Instance | $\geq 1/2$ [14]<br>$\leq 1-1/e$ [34] | $1/2 \to\geq$ **1 - 1/e** [3] [§7]<br>$\leq 0.826$ [40] |

Table 1: New competitive ratios are **bolded**. "$\geq$" refers to lower bounds on the *optimal* competitive ratio (algorithmic results), "$\leq$" refers to upper bounds (impossibility/hardness results), and arrows indicate improvement from the state of the art. "$-$" indicates that no constant competitive ratio is attainable, and "?" means that no previous competitive ratio was known.

**Known I.D. Instance.** We first consider the setting when the algorithm is given a type graph $H_{\text{typ}} = (U, B, F)$ and distributions $(\mathcal{D}_i)_{i=1}^n$, and executes on $G \sim (H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$. Observe that if $p_e \in \{0,1\}$ for each $e \in F$ of $H_{\text{typ}} = (U, B, F)$, then probing is unnecessary, and $\mathbb{E}[\text{OPT}(G)]$ corresponds 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 the problem generalizes either the **prophet inequality matching problem** or the **prophet secretary matching problem**, depending on if we work with adversarial arrivals or random order arrivals, respectively.

**Adversarial Order Arrivals.** In the AOM, we attain a competitive ratio of 1/2. This is a tight bound since the problem generalizes the classical single item prophet inequality for which 1/2 is an optimal competitive ratio [36]. Brubach et al. [11] independently proved the same competitive ratio against the weaker *online* adaptive benchmark (see the definition in Subsection 2.1). Our results are incomparable, as their result can be applied to an *unknown* patience framework (at a loss in competitive ratio), whereas our result holds against a stronger benchmark. Note that our result also holds assuming *known* downward-closed online probing constraints, as we discuss in Subsection 3.1.

**Random Order Arrivals.** In the ROM, we attain a competitive ratio of $1-1/e$. 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 [6, 18]). Huang et al. [32] very recently proved a 0.703 hardness result for multiple offline nodes and known i.i.d. arrivals.

**Known I.I.D. Instance.** The special case of identical distributions has been studied in multiple works [7, 3, 15, 16], beginning with the 0.12 competitive ratio of Bansal et al. [7], and which the previous best competitive ratio of 0.46 is due to Brubach et al. [16]. Fata et al. [24] improved this competitive ratio to 0.51 for the special case of **unbounded patience** (i.e., $\ell_b = \infty$ for all $b \in B$). All of these previous competitive ratios (including ours) are proven against the offline adaptive benchmark. Our $1-1/e$ competitive ratio for a known i.d. instance and random order arrivals improves on both of these bounds, while simultaneously applying to non-identical distributions. Note that Brubach et al. [13, 11] independently achieved a $1-1/e$ competitive ratio for a known i.i.d. instance, however their ratio is again proven against the weaker online adaptive benchmark. Our $1-1/e$ result matches the previously best known competitive ratio when probing is not required due to Manshadi et al. [40]. Very recently, Yan [49] improved on [40] and showed

[3]This competitive ratio holds when $G$ is unweighted, and for certain cases when $G$ is vertex-weighted. We defer the precise statement of the result to Theorem 7.7 of Section 7.

that $0.645 > 1 - 1/e$ is attainable when probing is not required.

**Known Stochastic Graph Instance.** An important case of the online stochastic matching problem with a known i.d. instance is the case of a **known stochastic graph**. In this setting, $H_{\text{typ}} = (V, B, F)$ 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 $H_{\text{typ}}$. 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 operates in the probe-commit model, and respects the patience values $(\ell_v)_{v \in V}$ as well as the ordering on $V$.

As described in Section 1, the focus of the original *offline* stochastic matching problem is to design efficient offline probing algorithms which attain approximation ratios against the offline adaptive benchmark (see [17, 1, 7, 3, 8, 26, 12, 43] for a collection of such results). Since all our probing algorithms are efficient, one of the main benefits of working with the offline adaptive benchmark opposed to the *online* adaptive benchmark, is that our competitive ratios imply approximation ratios. In particular, we get a $1 - 1/e$ approximation ratio for stochastic graphs with one-sided patience values. For context, 0.426 is the previously best known approximation ratio for bipartite graphs with one-sided patience values due to Pollner et al. [43]. Note that their algorithm has the benefit of working for random order *edge arrivals*, whereas ours requires one-sided random order vertex arrivals.

Gamlath et al. [26] also consider an offline bipartite matching problem for the special case when $G = (U, V, E)$ has unbounded patience, which they refer to as the **query-commit problem**. Both of our algorithms process $V$ in random order, and attain performance guarantees of $1 - 1/e$ against very different non-standard LPs (linear programs) – 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 A 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 unbounded patience setting the LPs take on the same value, as we prove in Proposition A.2 of Appendix A. Thus, our $1 - 1/e$ competitive ratio can be viewed as a generalization of their work to arbitrary patience values and more general probing constraints, as well as to known i.d. instances with random order arrivals. For the special case when $G$ is known and has unbounded patience, Derakhshan and Farhadi [20] recently proved an approximation ratio of $1 - 1/e + \delta$ for $\delta \geq 0.0014$.

Our $1 - 1/e$ approximation ratio is attained by a **non-adaptive** probing algorithm. An offline probing algorithm is non-adaptive on $G$ if the edges it probes (and the order this is done in) can be described as a randomized function of $G$. Equivalently, the edges probed by the algorithm are determined independently from the edge states $(\text{st}(e))_{e \in E}$ of $G$. In Theorem 9.1 of Section 9, we prove a $1 - 1/e$ hardness result against OPT which applies to *all* non-adaptive probing algorithms. Thus, we show that the **adaptivity gap** of the offline stochastic matching problem with one-sided patience values is exactly $1 - 1/e$ (see Corollary 9.4 for details).

**Edge-weighted Worst-case Instance.** We next consider the online stochastic matching problem in the most challenging setting for the algorithm. That is, the stochastic graph $G = (U, V, E)$ has edge weights and is generated by an adversary. When the edge probabilities of $G$ are binary (i.e., $p_e \in \{0, 1\}$ for all $e \in E$), this is the edge weighted online bipartite matching problem. It is well known that no constant competitive ratio is attainable for adversarial arrivals, and so we instead consider random order arrivals. This is the **secretary matching problem** in which Kesselheim et al. [35] proved an optimal asymptotic competitive ratio of $1/e$ (note that this is optimal even for a single offline vertex, i.e., $|U| = 1$). We generalize their matching algorithm so as to apply to the stochastic probing setting, and recover the same asymptotic $1/e$ competitive ratio.

**Vertex-weighted Worst-case Instance.** Purohit et al. [44] and Brubach et al. [14] both independently designed an *efficient* probing strategy, referred to as DP-OPT, which matches an online node $v$ to an edge of maximum weight in expectation, no matter the patience value $\ell_v$. Using DP-OPT, Brubach et al. considered a deterministic *greedy* probing algorithm when $G$ has offline vertex weights $(w_u)_{u \in U}$ and is a worst-case instance. In this setting, they prove a competitive ratio of $1/2$ in the AOM, and this is best possible amongst *deterministic* online probing algorithms.

When $G$ is unweighted, this greedy probing algorithm is very simple as it probes the available edges of an arriving vertex $v \in V$ based on non-increasing edge probabilities. In this setting, we prove a competitive ratio of $1 - 1/e$ for random order arrivals. We then extend this competitive ratio to a number of important settings when $G$ is vertex-weighted, yet **rankable** (see Section 7 for the precise definition). **Rankability** includes when $G$ is unweighted, as well as the well-studied case when $G$ has unit patience values [41, 42, 33, 29]. In this latter setting, each $v \in V$ *ranks* its adjacent edges in non-increasing order of $(w_u \cdot p_{u,v})_{u \in U}$. Assuming adversarial arrivals, this is the *online matching with stochastic rewards problem*, as described in Section 1. Mehta and Panigrahi [41] showed that $0.621 < 1 - \frac{1}{e}$ is a randomized in-approximation for this problem with regard to guarantees made against LP-std-unit. Here LP-std-unit is the LP used in [41] to upper bound/relax the offline adaptive benchmark in the unit patience setting. Our $1 - 1/e$ result in fact holds against LP-std-unit which implies that deterministic probing algorithms in the ROM have strictly more power than randomized probing algorithms in the AOM (see Corollary 7.9 for details). This contrasts with the classical online bipartite matching setting where such a separation is *not* known. We also prove that the same algorithm attains an *asymptotic* (as $|G| \to \infty$) competitive ratio of $1 - 1/e$ when $p_v := \max_{e \in \partial(v)} p_e$ satisfies $p_v = o(\ell_v)$ for each $v \in V$. The vanishing probability setting is similar in spirit to the small bid to budget assumption in the Adwords problem (see Goyal and Udwani [29] for details).

Finally, we generalize DP-OPT to downward-closed probing constraints and extend our competitive ratios in a number of settings. Note that DP-OPT is crucial for ensuring the efficiency of our other online probing algorithms, as we explain in Section 8.

## 3.1 Generalizing to Downward Closed Online Probing Constraints

Consider an online vertex $v \in V$ of a stochastic graph $G = (U, V, E)$. The patience parameter $\ell_v$ can be viewed as a simple budgetary constraint, where each probe has unit cost and the patience parameter is the budget. A natural generalization is thus to instead consider a **knapsack** (or **budgetary**) constraint. That is, a non-negative **budget** $B_v \geq 0$ and **edge probing costs** $(c_e)_{e \in \partial(v)}$, such that any subset $S \subseteq \partial(v)$ may be probed (in any order), provided $\sum_{e \in S} c_e \leq B_v$. Alternatively, $\ell_v$ can be viewed as an $\ell_v$-uniform-matroid constraint on $\partial(v)$, and so an incomparable generalization is to an *arbitrary* matroid constraint on $\partial(v)$. Matroid probing constraints of the latter form were considered in the offline setting in [30, 4]. We introduce a general online probing framework that encompasses both budgetary and matroid probing constraints.

Define $\partial(v)^{(*)}$ as the collection of strings (tuples) formed from the edges of $\partial(v)$ whose characters (entries) are all distinct. Note that we use string/tuple notation and terminology interchangeably. Each $v \in V$ has an **(online) probing constraint** $\mathcal{C}_v \subseteq \partial(v)^{(*)}$ which we assume is **downward-closed**. That is, $\mathcal{C}_v$ has the property that if $\boldsymbol{e} \in \mathcal{C}_v$, then so is any substring or permutation of $\boldsymbol{e}$. Clearly our setting encodes the case when $v$ has a patience value $\ell_v$, and more generally, when $\mathcal{C}_v$ corresponds to a matroid or budgetary constraint.

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 $\boldsymbol{e} \in \partial(v)^{(*)}$ for $v \in V$, thus determining in a single operation whether or not $\boldsymbol{e} \in \partial(v)^{(*)}$ is in $\mathcal{C}_v$. Assuming access to such an oracle, all our algorithms are implementable in polynomial time, as we prove in Section 8. When a probing constraint can be stated explicitly, our algorithms are polynomial time in the usual (e.g., Turing machine time) sense. For example, for a budget constraint, the budget $B_v$ and edge probing costs $(c_e)_{e \in \partial(v)}$ are revealed upon the arrival of the online vertex $v$.

For the remainder of the paper, when discussing a stochastic graph $G = (U, V, E)$, we work in the full generality of online probing constraints $(\mathcal{C}_v)_{v \in V}$, unless indicated otherwise.

## 3.2 An Overview of Our Techniques

We first describe our techniques in the known stochastic graph setting (i.e, when the stochastic graph $G = (U, V, E)$ is edge-weighted and known to the online probing algorithm). Afterwards, we explain how our techniques apply when $G$ is unknown and drawn from a known i.d. input or generated by an adversary. We conclude by describing our techniques when $G$ is vertex-weighted.

The main challenge in designing *efficient* probing algorithms (whether offline or online), is that in general it is infeasible to efficiently compute the decisions made by the offline adaptive benchmark (i.e., OPT). One of the standard approaches in the literature is to instead relax OPT via an LP. When $G$ has patience values $(\ell_v)_{v \in V}$, Bansal et al. [7] introduced an LP which assigns fractional values to the edges of $G$, say $(x_e)_{e \in E}$, such that $x_e$ upper bounds the probability $e$ is probed by OPT. 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. [27]), one can round the values $(x_e)_{e \in \partial(v)}$ to determine $\ell_v$ 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 [7, 3, 8, 16, 12, 43]). 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 $\text{OPT}(G)$ and an optimum solution to this LP can be as small as 0.51, so our $1 - 1/e$ competitive ratio cannot be achieved via a comparison to this LP, even for the special case when $G$ is known and has patience values.

**Defining LP-config:** Our approach is to work with a new configuration LP (LP-config) which we can state even when $G$ has online probing constraints $(\mathcal{C}_v)_{v \in V}$. 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 $\lambda$. 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 $\text{val}(\boldsymbol{e}) := \sum_{f \in \boldsymbol{e}} w_f \cdot p_f \cdot q(\boldsymbol{e}_{<f})$ corresponds to the expected weight of the first active edge revealed if $\boldsymbol{e}$ is probed in order of its entries. For each $v \in V$ and $\boldsymbol{e} \in \mathcal{C}_v$, we introduce a decision variable

$x_v(\boldsymbol{e})$ and state the following LP:

$$\begin{aligned}
&\text{maximize} && \sum_{v \in V} \sum_{\boldsymbol{e} \in \mathcal{C}_v} \text{val}(\boldsymbol{e}) \cdot x_v(\boldsymbol{e}) && && \text{(LP-config)}\\
&\text{subject to} && \sum_{v \in V} \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_v: \\ (u,v) \in \boldsymbol{e}}} p_{u,v} \cdot q(\boldsymbol{e}_{<(u,v)}) \cdot x_v(\boldsymbol{e}) \leq 1 && \forall u \in U && (3.1)\\
& && \sum_{\boldsymbol{e} \in \mathcal{C}_v} x_v(\boldsymbol{e}) = 1 && \forall v \in V, && (3.2)\\
& && x_v(\boldsymbol{e}) \geq 0 && \forall v \in V, \boldsymbol{e} \in \mathcal{C}_v && (3.3)
\end{aligned}$$

We first prove that LP-config is a **relaxation** of the offline adaptive benchmark. 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 3.1.** *$OPT(G) \leq LPOPT(G)$.*

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 [45, 28]. The separation oracle we require is precisely the greedy probing strategy we analyze in the vertex-weighted setting of Section 7. We provide the details of this reduction in Section 8.

For the case of patience values, a closely related LP was independently introduced by Brubach et al. [13, 11] to design probing algorithms for known i.i.d. instances and known i.d. instances with adversarial arrivals. Their competitive ratios are proven against an optimal solution to this LP. They then argue that the LP solution relaxes the *online* adaptive benchmark, which is a weaker statement than Theorem 3.1. We explain below why this weaker statement is easier to prove.

**Proving Theorem 3.1:** In order to prove Theorem 3.1, one 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 first *hypothetically* assume that the following restrictive assumptions on the offline adaptive benchmark hold without loss of generality (w.l.o.g.):

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

($P_2$) For each $v \in V$, the edge probes involving $\partial(v)$ are determined 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 (3.1) 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 $\text{LPOPT}(G)$. Unfortunately, while we can assume ($P_1$) holds w.l.o.g., we cannot simultaneously assume ($P_2$). This is because 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 $v$. An alternative approach would be to define $x_v(\boldsymbol{e})$ as the probability that the offline adaptive benchmark probes $\boldsymbol{e}$ in order, *conditioned* on its first $|\boldsymbol{e}| - 1$ edges being inactive. In this case, $\text{OPT}(G) = \sum_{v \in V} \sum_{\boldsymbol{e} \in \mathcal{C}_v} \text{val}(\boldsymbol{e}) \cdot x_v(\boldsymbol{e})$; however then $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$ need not satisfy constraint (3.2). This is because for a fixed $v \in V$, $\sum_{\boldsymbol{e} \in \mathcal{C}_v} x_v(\boldsymbol{e})$ is a sum over $|\mathcal{C}_v|$ *conditional* probabilities – each of which conditions on a *different* event – and so its value could exceed 1. Thus, neither of these decision variable interpretations seems to lead to an easy proof of Theorem 3.1.

Before continuing, we note that when working with the *online* adaptive benchmark of Brubach et al. [11, 13], $(P_1)$ and $(P_2)$ *can* be assumed to hold w.l.o.g. As a result of the discussion following $(P_1)$ and $(P_2)$, this explains why Brubach et al. are able to easily argue that $\text{LPOPT}(G)$ upper bounds this weaker benchmark on $G$.

Returning to the proof of Theorem 3.1, our solution is to introduce an alternative interpretation of $(x_v(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$ based upon a *relaxation* of the offline stochastic matching problem we refer to as the **relaxed stochastic matching problem**. A solution to this problem is a **relaxed probing algorithm**. 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*. Observe that if $\text{OPT}_{\text{rel}}(G)$ is the performance of an *optimal* relaxed probing algorithm, then by definition $\text{OPT}(G) \leq \text{OPT}_{\text{rel}}(G)$. Crucially, there exists an optimal relaxed probing algorithm which satisfies $(P_1)$ and which is **non-adaptive**. A relaxed probing algorithm is said to be non-adaptive, provided its edge probes are determined independently of the edge states $(\text{st}(e))_{e \in E}$. Non-adaptivity is a much stronger property than $(P_2)$, and so by the above discussion we are able to conclude that $\text{OPT}_{\text{rel}}(G) \leq \text{LPOPT}(G)$. Since $\text{OPT}(G) \leq \text{OPT}_{\text{rel}}(G)$, this implies Theorem 3.1. 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 4.1 of Section 4. Note that there may be a simpler proof of Theorem 3.1, however our relaxed stochastic matching problem exactly characterizes LP-config (i.e., $\text{OPT}_{\text{rel}}(G) = \text{LPOPT}(G)$), and so it helps us understand LP-config. For instance, in Appendix A, we show that in the unbounded patience setting, LP-QC of [26] 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

$$\widetilde{x}_e := \sum_{\substack{\boldsymbol{e}' \in \mathcal{C}_v: \\ e \in \boldsymbol{e}'}} q(\boldsymbol{e}'_{<e}) \cdot x_v(\boldsymbol{e}'). \tag{3.4}$$

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 (3.2) 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 3.2.** *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 3.3.** We can view Lemma 3.2 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 unbounded patience setting (i.e., $\mathcal{C}_v = \partial(s)^{(*)}$), and the procedure is much more complicated than ours (see Theorem A.3 of Appendix A for details).

**Definition 1** (Propose - Known Stochastic Graph)**.** We say that `VertexProbe` **proposes** $s$ to the vertex $u \in N(s)$ provided the algorithm outputs $(u, s)$ when executing on the fixed node $s \in V$. When it is clear that `VertexProbe` is being executed on $s$, we say that $s$ proposes to $u$.

Consider now the following online probing algorithm, where the ordering $\pi$ on $V$ is either adversarial or u.a.r.

**Algorithm 1** Known Stochastic Graph

**Input:** a stochastic graph $G = (U, V, E)$.
**Output:** 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 $\pi$ 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: return $\mathcal{M}$.
```

**Remark 3.4.** 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 when $\pi$ is adversarial, and its competitive ratio is only $1/2$ when $\pi$ is u.a.r. Thus, we must modify the algorithm to prove $1/2$ and $1 - 1/e$ competitive ratios, even in the known stochastic graph setting. When $\pi$ is adversarial, we reduce the problem to designing a $1/2$-**selectable online contention resolution scheme (OCRS)** for a 1-uniform matroid [4]. When $\pi$ is u.a.r., we reduce the problem to designing a $1 - 1/e$-selectable **random order contention resolution scheme (RCRS)** for a 1-uniform matroid. (See Section 5 for formal definitions). In both cases, such contention resolution schemes are known to exist (see [23] for the OCRS, and [37] for the RCRS). We now provide a high level overview of the reduction for the known stochastic graph case. Each CRS we use is for a 1-uniform matroid, so we omit this in the descriptions below.

First observe that $\text{LPOPT}(G) = \sum_{e \in E} w_e p_e \widetilde{x}_e$, and $\text{OPT}(G) \leq \text{LPOPT}(G)$ by Theorem 3.1. Thus, it suffices to design an online probing algorithm which matches each $e \in E$ with probability $1/2 \cdot z_e$ when $\pi$ is generated by an adversary, and $(1 - 1/e) \cdot z_e$ when $\pi$ is generated u.a.r, where $z_e := p_e \widetilde{x}_e$. Observe that for a fixed $u \in U$, as Algorithm 1 executes, each $v \in V$ proposes to $u$ independently with probability $z_{u,v}$. On the other hand, $\sum_{v \in V} z_{u,v} \leq 1$ by (3.1) of LP-config. Let us first assume $\pi$ is adversarial. Observe that if $u$ executes its own $\alpha$-selectable OCRS on ground set $\partial(u)$ as Algorithm 1 executes, then each $e \in \partial(u)$ will be matched to $u$ with probability $\alpha \cdot z_e$. By having *each* $u \in U$ concurrently execute its own $\alpha$-selectable OCRS, the resulting probing algorithm will be $\alpha$-competitive. Thus, an $\alpha$-selectable OCRS can be used to design an $\alpha$-competitive online probing algorithm in the AOM. Similarly, when $\pi$ is u.a.r., an $\alpha$-selectable RCRS can be used to design an $\alpha$-competitive online probing algorithm in the ROM.

For random order arrivals, the RCRS based approach simplifies the pricing based approach Gamlath et al. [26] used to attain a competitive ratio of $1 - 1/e$ in the special case of unbounded patience. This simplified approach was also observed by Fu et al. [25] in the context of the Gamlath et al. LP (LP-QC). They assume unbounded patience in the probe-commit model, and

[4]The independent sets of a 1-uniform matroid on ground set $S$ are the subsets of $S$ of size at most 1.

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. For context, 0.395 is the best known competitive ratio when allowing for arbitrary patience values and random order edge arrivals [43] (this was recently improved to 0.474 in the unbounded patience setting [39]).

### 3.2.1 Extending to Known I.D. Instances:

Consider when $G$ is unknown and drawn from a known i.d. input $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$. In this case, we generalize 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 reduce to 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 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.

### 3.2.2 Worst-Case Instances and Random-Order Arrivals.

Suppose that the adversary chooses a graph $G = (U, V, E)$ whose online vertices arrive in random order $v_1, \ldots, v_n$. Upon receiving the online vertices $V_t := \{v_1, \ldots, v_t\}$, in order to generalize the matching algorithm of Kesselheim et al. [35], we would ideally probe the edges of $\partial(v_t)$ suggested by OPT on $G_t$, where $G_t := G[U \cup V_t]$ is the induced stochastic graph on $U \cup V_t$. However, since we wish for our algorithms to be efficient in addition to attaining optimal competitive ratios, this strategy is not feasible. We instead solve LP-config for $G_t$ to get a solution $(x_v(\boldsymbol{e}))_{v \in V_t, \boldsymbol{e} \in \mathcal{C}_v}$, and then execute `VertexProbe` on $(v_t, \partial(v_t), (x_v(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_{v_t}})$. If edge $e_t = (u_t, v_t)$ is returned, and $u_t$ is unmatched, then we add $e_t$ to the current matching. In Theorem 6.3 of Section 6, we show that this online algorithm attains an asymptotic competitive ratio of $1/e$. The analysis follows a similar proof structure as presented in [35].

### 3.2.3 Solving LP-config Efficiently and Vertex-weighted Worst-case Instances

Let $v \in V$ be a fixed vertex of $G = (U, V, E)$. In Theorem 7.1, we show how to probe $\partial(v)$ in such a way that $v$ is matched to a neighbouring edge $e = (u, v)$ whose weight $w_e$ is as large as possible in expectation. This is precisely the behaviour of OPT on the *induced stochastic graph* $G[\{v\} \cup U]$. While computing such a strategy is computationally challenging for a general graph, in Theorem 7.1 we show that it can be done efficiently for the *star graph* $G[\{v\} \cup U]$. This was first observed by Purohit et al. [44] and Brubach et al. [14] for the special case when $v$ has patience $\ell_v$. Using a similar dynamic programming (DP) based approach, we generalize their algorithm DP-OPT to apply to an arbitrary downward-closed constraint $\mathcal{C}_v$. In Theorem 8.1 of Section 8, we show that DP-OPT provides a separation oracle for the dual of LP-DP, thus leading to an efficient way to solve LP-config (despite it having exponentially many variables).

Suppose now that $G = (U, V, E)$ has vertex weights $(w_u)_{u \in U}$. In Section 7, we use DP-OPT to design a *greedy* online probing algorithm for $G$. Upon the arrival of $v$, if $R \subseteq U$ denotes the unmatched vertices available, we apply DP-OPT to determine which edges of $G[\{v\} \cup R]$ to probe. We analyze our algorithm using an LP relaxation of OPT called LP-DP, a generalization of an LP introduced by Brubach et al. [14] to arbitrary probing constraints $(\mathcal{C}_v)_{v \in V}$. For both adversarial and random order arrivals, we apply the primal-dual method of Devanur et al. [21]. As the matching of our algorithm is constructed, we simultaneously construct a (random) dual solution of LP-DP. For adversarial arrivals, the dual solution is feasible in expectation, and twice as large as the expected weight of the algorithm's matching, thus leading to a competitive ratio of $1/2$. When the vertices arrive in random order, we assign dual variables using $g(Y_v)$, where $g(y) := e^{y-1}$ is the function used to analysis the RANKING algorithm in [21], and $Y_v \in [0,1]$ is the *random arrival time* of $v$. Interestingly, the *non-monotonic* behaviour of our greedy algorithm prevents the dual-solution from being feasible in expectation, and so we are not able to prove an unconditional competitive ratio of $1 - 1/e$. Instead, in Theorem 7.7 we characterize a number of important inputs in which our algorithm is monotonic, thus allowing us to prove a conditional competitive ratio of $1 - 1/e$.

## 4 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

$$\text{OPT}(G) \leq \text{OPT}_{\text{rel}}(G). \tag{4.1}$$

We say that $\mathcal{A}$ is **non-adaptive**, provided its edge probes can be described as a (randomized) function of $G$. Equivalently, $\mathcal{A}$ is non-adaptive if the edges 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 4.1.** *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 4.2.** Note that ($Q_2$) implies the hypothetical property ($P_2$), yet is much stronger.

Let us assume that Lemma 4.1 holds for now.

*Proof of Theorem 3.1.* Consider algorithm $\mathcal{B}$ of Lemma 4.1, 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 (4.1), Theorem 3.1 follows. $\square$

**Remark 4.3.** As mentioned, LP-config is an exact LP formulation of the relaxed stochastic matching problem, as we prove in Theorem A.1 of Appendix A.

## 4.1 Proving Lemma 4.1

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 4.1, 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 4.1.

Suppose that $\boldsymbol{e} \in E^{(*)}$, and recall that $\lambda$ 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

$$\sum_{\substack{e \in \partial(v): \\ (\boldsymbol{e}', e) \in \mathcal{C}_v}} y_v(\boldsymbol{e}', e) \leq y_v(\boldsymbol{e}'), \tag{4.2}$$

for each $\boldsymbol{e}' \in \mathcal{C}_v$.

**Proposition 4.4.** *Given a collection of values $(y_v(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_v}$ which satisfy $y_v(\lambda) = 1$ and (4.2), 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*

$$\mathbb{P}[(\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_k) = (e_1, \ldots, e_k)] = y_v(\boldsymbol{e}), \tag{4.3}$$

*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 Proposition 4.4.* First define $\mathcal{C}_v^{>} := \{\boldsymbol{e}' \in \mathcal{C}_v : y_v(\boldsymbol{e}') > 0\}$, which we observe is downward-closed since by assumption $\mathcal{C}_v$ is downward-closed and (4.2) holds. We prove the proposition for $\mathcal{C}_v^{>}$, which we then argue implies the proposition holds for $\mathcal{C}_v$. Observe now that for each $\boldsymbol{e}' \in \mathcal{C}_v^{>}$, we have that

$$\sum_{\substack{e \in \partial(v): \\ (\boldsymbol{e}', e) \in \mathcal{C}_v^{>}}} \frac{y_v(\boldsymbol{e}', e)}{y_v(\boldsymbol{e}')} \leq 1 \tag{4.4}$$

as a result of (4.2) (recall that $y_v(\lambda) := 1$). We thus define for each $\boldsymbol{e}' \in \mathcal{C}_v^{>}$,

$$z_v(\boldsymbol{e}') := 1 - \sum_{\substack{e \in \partial(v): \\ (\boldsymbol{e}', e) \in \mathcal{C}_v^{>}}} \frac{y_v(\boldsymbol{e}', e)}{y_v(\boldsymbol{e}')}, \tag{4.5}$$

which we observe has the property that $0 \leq z_v(\boldsymbol{e}') < 1$. The strict inequality follows from the definition of $\mathcal{C}_v^{>}$. This leads to the following randomized rounding algorithm, which we claim outputs a random string $\boldsymbol{Y}$ which satisfies the desired properties:

**Algorithm 2** VertexRound

**Input:** a collection of values $(y_v(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_v^{>}}$ satisfying (4.2) and $y_v(\lambda) = 1$.
**Output:** a random string $\boldsymbol{Y} = (Y_1, \ldots, Y_{|\partial(v)|})$ supported on $\mathcal{C}_v^{>}$.

```
1: Set $\boldsymbol{e}' \leftarrow \lambda$.
2: Initialize $Y_i = \lambda$ for each $i = 1, \ldots, |\partial(v)|$.
3: for $i = 1, \ldots, |\partial(v)|$ do
4:     Exit the "for loop" with probability $z_v(\boldsymbol{e}')$.    ▷ pass with a certain probability – see (4.5)
5:     Draw $e \in \partial(v)$ satisfying $(\boldsymbol{e}', e) \in \mathcal{C}_v^{>}$ with probability $y_v(\boldsymbol{e}', e)/(y_v(\boldsymbol{e}')\,(1 - z_v(\boldsymbol{e}')))$.
6:     Set $Y_i = e$.
7:     $\boldsymbol{e}' \leftarrow (\boldsymbol{e}', e)$.
8: return $\boldsymbol{Y} = (Y_1, \ldots, Y_{|\partial(v)|})$.  ▷ concatenate the edges in order and return the resulting string
```

Clearly, the random string $\boldsymbol{Y}$ is supported on $\mathcal{C}_v^{>}$, thanks to line 5 of Algorithm 2. We now show that (4.3) holds. As such, let us first assume $k = 1$, and $e \in \partial(v)$ satisfies $(e) \in \mathcal{C}_v^{>}$. Observe that

$$\mathbb{P}[Y_1 = e] = (1 - z_v(\lambda)) \frac{y_v(e)}{1 - z_v(\lambda)} = y_v(e),$$

as the algorithm does not exit the "for loop" with probability $1 - z_v(\lambda)$, in which case it draws $e$ with probability $y_v(e)/(1 - z_v(\lambda))$. In general, take $k \geq 2$, and assume that for each $\boldsymbol{e}' \in \mathcal{C}_v^{>}$ with $1 \leq |\boldsymbol{e}'| < k$, it holds that $\mathbb{P}[(Y_1, \ldots, Y_k) = \boldsymbol{e}'] = y_v(\boldsymbol{e}')$. If we now fix $\boldsymbol{e} = (e_1, \ldots, e_k) \in \mathcal{C}_v^{>}$ with $|\boldsymbol{e}| = k$, observe that $\boldsymbol{e}_{<k} := (e_1, \ldots, e_{k-1}) \in \mathcal{C}_v^{>}$, as $\mathcal{C}_v^{>}$ is downward-closed. Moreover,

$$\begin{aligned}\mathbb{P}[(Y_1, \ldots, Y_k) = \boldsymbol{e}] &= \mathbb{P}[Y_k = e_k \,|\, (Y_1, \ldots, Y_{k-1}) = \boldsymbol{e}_{<k}] \cdot \mathbb{P}[(Y_1, \ldots, Y_{k-1}) = \boldsymbol{e}_{<k}] \\ &= \mathbb{P}[Y_k = e_k \,|\, (Y_1, \ldots, Y_{k-1}) = \boldsymbol{e}_{<k})] \cdot y_v(\boldsymbol{e}_{<k}),\end{aligned}$$

where the last line follows by the induction hypothesis since $\boldsymbol{e}_{<k} \in \mathcal{C}_v^{>}$ is of length $k-1$. We know however that

$$\mathbb{P}[Y_k = \boldsymbol{e}_k \,|\, (Y_1, \ldots, Y_{k-1}) = \boldsymbol{e}_{<k}] = (1 - z_v(\boldsymbol{e}_{<k}))\, \frac{y_v(\boldsymbol{e}_{<k}, e_k)}{y_v(\boldsymbol{e}_{<k})\,(1 - z_v(\boldsymbol{e}_{<k}))} = \frac{y_v(\boldsymbol{e}_{<k}, e_k)}{y_v(\boldsymbol{e}_{<k})}.$$

This is because once we condition on the event $(Y_1, \ldots, Y_{k-1}) = \boldsymbol{e}_{<k}$, we know that the algorithm does not exit the "for loop" with probability $1 - z_v(\boldsymbol{e}_{<k})$, in which case it selects $e_k \in \partial(v)$ with probability $y_v(\boldsymbol{e}_{<k}, e_k)/(y_v(\boldsymbol{e}_{<k})\,(1 - z_v(\boldsymbol{e}_{<k})))$, since $(\boldsymbol{e}_{<k}, e_k) \in \mathcal{C}_v^{>}$ by assumption. As such, we have that $\mathbb{P}[(Y_1, \ldots, Y_k) = \boldsymbol{e}] = y_v(\boldsymbol{e})$, and so the proposition holds for $\mathcal{C}_v^{>}$. To complete the argument, observe that since $\boldsymbol{Y}$ is supported on $\mathcal{C}_v^{>}$, the substrings of $\boldsymbol{Y}$ are also supported on $\mathcal{C}_v^{>}$, as $\mathcal{C}_v^{>}$ is downward-closed. Thus, $\boldsymbol{Y}$ satisfies (4.3) for the non-empty strings of $\mathcal{C}_v \setminus \mathcal{C}_v^{>}$, in addition to the non-empty strings of $\mathcal{C}_v^{>}$. ☐

*Proof of Lemma 4.1.* 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 $\text{st}(e) = 1$, 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 $\lambda$ 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

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

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

$$\mathbb{P}[S(\boldsymbol{e}) \cap \{\boldsymbol{X}_{\leq k}^v = \boldsymbol{e}\}] = \mathbb{P}[\{\text{st}(e_k) = 1\} \cap \{\boldsymbol{X}_{\leq k}^v = \boldsymbol{e}\}], \tag{4.6}$$

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

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

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

$$\mathbb{E}[w(\mathcal{M})] = \sum_{v \in V} \sum_{\substack{\boldsymbol{e}=(e_1,\ldots,e_k) \in \mathcal{C}_v: \\ k \geq 1}} w_{e_k} p_{e_k} \mathbb{P}[\boldsymbol{X}_{\leq k}^v = \boldsymbol{e}]. \tag{4.7}$$

Our goal is to find a non-adaptive relaxed probing algorithm which matches the value of (4.7). Thus, for each $v \in V$ and $\boldsymbol{e} = (e_1, \ldots, e_k) \in \mathcal{C}_v$ with $k \geq 1$, define $x_v(\boldsymbol{e}) := \mathbb{P}[\boldsymbol{X}_{\leq k}^v = \boldsymbol{e}]$, where $x_v(\lambda) := 1$. Observe now that for each $\boldsymbol{e}' = (e_1', \ldots, e_k') \in \mathcal{C}_v$,

$$\sum_{\substack{e \in \partial(v): \\ (\boldsymbol{e}', e) \in \mathcal{C}_v}} \mathbb{P}[\boldsymbol{X}_{\leq k+1}^v = (\boldsymbol{e}', e) \mid \boldsymbol{X}_{\leq k}^v = \boldsymbol{e}'] \leq 1 - p_{e_k'}. \tag{4.8}$$

To see (4.8), 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 (4.8). Using (4.8), the values $(x_v(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_v}$ satisfy

$$\sum_{\substack{e \in \partial(v): \\ (\boldsymbol{e}', e) \in \mathcal{C}_v}} x_v(\boldsymbol{e}', e) \leq (1 - p_{e'_k}) \cdot x_v(\boldsymbol{e}'), \tag{4.9}$$

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 $\boldsymbol{e} \in \mathcal{C}_v$

$$y_v(\boldsymbol{e}) := \begin{cases} x_v(\boldsymbol{e})/q(\boldsymbol{e}_{<|\boldsymbol{e}|}) & \text{if } |\boldsymbol{e}| \geq 1, \\ 1 & \text{otherwise.} \end{cases} \tag{4.10}$$

Observe that (4.9) ensures that for each $\boldsymbol{e}' \in \mathcal{C}_v$,

$$\sum_{\substack{e \in \partial(v): \\ (\boldsymbol{e}', e) \in \mathcal{C}_v}} y_v(\boldsymbol{e}', e) \leq y_v(\boldsymbol{e}'), \tag{4.11}$$

and $y_v(\lambda) := 1$. As a result, Proposition 4.4 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 $|\boldsymbol{e}| = k \geq 1$,

$$\mathbb{P}[\boldsymbol{Y}^v_{\leq k} = \boldsymbol{e}] = y_v(\boldsymbol{e}). \tag{4.12}$$

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 4.1:

**Algorithm 3** Algorithm $\mathcal{B}$

```
Input: a stochastic graph $G = (U, V, E)$.
Output: a one-sided matching $\mathcal{N}$ of $G$ of active edges.
1: Set $\mathcal{N} \leftarrow \emptyset$.
2: Draw $(\boldsymbol{Y}^v)_{v \in V}$ according to the product distribution $\prod_{v \in V} \mathcal{D}^v$.
3: for $v \in V$ do
4:     for $i = 1, \ldots, |\boldsymbol{Y}^v|$ do
5:         Set $e \leftarrow \boldsymbol{Y}^v_i$.                ▷ $\boldsymbol{Y}^v_i$ is the $i^{th}$ edge of $\boldsymbol{Y}^v$
6:         Probe the edge $e$, revealing st$(e)$.
7:         if st$(e) = 1$ and $v$ is unmatched by $\mathcal{N}$ then
8:             Add $e$ to $\mathcal{N}$.
9: return $\mathcal{N}$.
```

Using (4.12) 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 4.1. $\square$

## 5 Known I.D. Instance Model

Suppose that $(H_{\mathrm{typ}}, (\mathcal{D}_i)_{i=1}^n)$ is a known i.d. input, where $H_{\mathrm{typ}} = (U, B, F)$ has downward-closed online probing constraints $(\mathcal{C}_b)_{b \in B}$. If $G \sim (H_{\mathrm{typ}}, (\mathcal{D}_i)_{i=1}^n)$, 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.

$$\text{maximize} \quad \sum_{i \in [n], b \in B, \boldsymbol{e} \in \mathcal{C}_b} \mathrm{val}(\boldsymbol{e}) \cdot x_i(\boldsymbol{e} \,||\, b) \qquad \text{(LP-config-id)}$$

$$\text{subject to} \quad \sum_{i \in [n], b \in B} \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_b: \\ (u,b) \in \boldsymbol{e}}} p_{u,b} \cdot q(\boldsymbol{e}_{<(u,b)}) \cdot x_i(\boldsymbol{e} \,||\, b) \leq 1 \qquad \forall u \in U \qquad (5.1)$$

$$\sum_{\boldsymbol{e} \in \mathcal{C}_b} x_i(\boldsymbol{e} \,||\, b) = r_i(b) \qquad \forall b \in B, i \in [n] \qquad (5.2)$$

$$x_i(\boldsymbol{e} \,||\, b) \geq 0 \qquad \forall b \in B, \boldsymbol{e} \in \mathcal{C}_b, i \in [n] \qquad (5.3)$$

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

One way to prove Theorem 5.1 is to use the properties of the relaxed benchmark on $G$ guaranteed by Lemma 4.1, 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 (5.1) as saying that the relaxed benchmark matches each offline vertex at most once in expectation. Moreover, (5.2) 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 below. Specifically, we consider an optimum solution of LP-config with respect to $G$, and apply a conditioning argument in conjunction with Theorem 3.1.

*Proof of Theorem 5.1.* Suppose that $(H_{\mathrm{typ}}, (\mathcal{D}_t)_{t=1}^n)$ is a known i.d. input, where $H_{\mathrm{typ}} = (U, B, F)$. Recall that $\mathcal{C}_b$ corresponds to the online probing constraint of each type node $b \in B$. For convenience, we denote $\mathcal{I} := \sqcup_{b \in B} \mathcal{C}_b$. We can then define the following collection of random variables, denoted $(X_t(\boldsymbol{e}))_{t \in [n], \boldsymbol{e} \in \mathcal{I}}$, based on the following randomized procedure:

- Draw the stochastic graph $G \sim (H_{\mathrm{typ}}, (\mathcal{D}_t)_{t=1}^n)$, whose vertex arrivals we denote by $v_1, \ldots, v_n$.
- Compute an optimum solution of LP-config for $G$, which we denote by $(x_{v_t}(\boldsymbol{e}))_{t \in [n], \boldsymbol{e} \in \mathcal{C}_{v_t}}$.
- For each $t = 1, \ldots, n$ and $\boldsymbol{e} \in \mathcal{I}$, set $X_t(\boldsymbol{e}) = x_{v_t}(\boldsymbol{e})$ if $\boldsymbol{e} \in \mathcal{C}_{v_t}$, otherwise set $X_t(\boldsymbol{e}) = 0$.

Observe then that since by definition $(X_{v_t}(\boldsymbol{e}))_{t \in [n], \boldsymbol{e} \in \mathcal{C}_{v_t}}$ is a feasible solution to LP-config for $G$, it holds that for each $t = 1, \ldots, n$

$$\sum_{\boldsymbol{e} \in \mathcal{I}} X_t(\boldsymbol{e}) = 1, \tag{5.4}$$

and

$$\sum_{t \in [n], b \in B} \sum_{\substack{\boldsymbol{e} \in \mathcal{I}: \\ (u,b) \in \boldsymbol{e}}} p_{u,b} \cdot q(\boldsymbol{e}_{<(u,b)}) \cdot X_t(\boldsymbol{e}) \leq 1, \tag{5.5}$$

for each $u \in U$. Moreover, $(X_t(\boldsymbol{e}))_{t \in [n], \boldsymbol{e} \in \mathcal{C}_{v_t}}$ is a optimum solution to LP-config for $G$, so Theorem 3.1 implies that

$$\text{OPT}(G) \leq \text{LPOPT}(G) = \sum_{t=1}^{n} \sum_{\boldsymbol{e} \in \mathcal{I}} \text{val}(\boldsymbol{e}) \cdot X_t(\boldsymbol{e}). \tag{5.6}$$

In order to make use of these inequalities in the context of the type graph $H_{\text{typ}}$, let us first fix a type node $b \in B$ and a string $\boldsymbol{e} \in \mathcal{C}_b$. For each $t \in [n]$, we can then define

$$x_t(\boldsymbol{e} \,||\, b) := \mathbb{E}[X_t(\boldsymbol{e}) \cdot \mathbf{1}_{[v_t = b]}], \tag{5.7}$$

where the randomness is over the generation of $G$. Observe that by definition of the $(X_t(\boldsymbol{e}))_{t \in [n], \boldsymbol{e} \in \mathcal{I}}$ values, $x_t(\boldsymbol{e} \,||\, b) = 0$, provided $\boldsymbol{e} \notin \mathcal{C}_b$. We claim that $(x_t(\boldsymbol{e} \,||\, b))_{b \in B, t \in [n], \boldsymbol{e} \in \mathcal{C}_b}$ is a feasible solution to LP-config-id. To see this, first observe that if we multiply (5.4) by the indicator random variable $\mathbf{1}_{[v_t = b]}$, then we get that $\sum_{\boldsymbol{e} \in \mathcal{I}} X_t(\boldsymbol{e}) \cdot \mathbf{1}_{[v_t = b]} = \mathbf{1}_{[v_t = b]}$. As a result, if we take expectations over this equality,

$$\sum_{\boldsymbol{e} \in \mathcal{I}} x_t(\boldsymbol{e} \,||\, b) = \sum_{\boldsymbol{e} \in \mathcal{I}} \mathbb{E}\left[X_t(\boldsymbol{e}) \cdot \mathbf{1}_{[v_t = b]}\right] = \mathbb{P}[v_t = b] =: r_t(b),$$

for each $b \in B$ and $t \in [n]$. Let us now fix $u \in U$. Observe that since $X_t(\boldsymbol{e}) \cdot \mathbf{1}_{[v_t = b]} = X_t(\boldsymbol{e})$ for each $\boldsymbol{e} \in \mathcal{C}_b$, (5.5) ensures that

$$\sum_{t \in [n], b \in B} \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_b: \\ (u,b) \in \boldsymbol{e}}} p_{u,b} \cdot q(\boldsymbol{e}_{<(u,b)}) \cdot X_t(\boldsymbol{e}) \cdot \mathbf{1}_{[v_t = b]} = \sum_{t \in [n], b \in B} \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_b: \\ (u,b) \in \boldsymbol{e}}} p_{u,b} \cdot q(\boldsymbol{e}_{<(u,b)}) \cdot X_t(\boldsymbol{e}) \leq 1 \tag{5.8}$$

Thus, after taking expectations over (5.8), $\sum_{t \in [n], b \in B} \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_b: \\ (u,b) \in \boldsymbol{e}}} p_{u,b} \cdot q(\boldsymbol{e}_{<(u,b)}) \cdot x_t(\boldsymbol{e} \,||\, b) \leq 1$, for each $u \in U$. Since $(x_t(\boldsymbol{e} \,||\, b))_{t \in [n], b \in B, \boldsymbol{e} \in \mathcal{C}_b}$ satisfies these inequalities, and the variables are clearly all non-negative, it follows that $(x_t(\boldsymbol{e} \,||\, b))_{t \in [n], b \in B, \boldsymbol{e} \in \mathcal{C}_b}$ is a feasible solution to LP-config-id. Let us now express the right-hand side of (5.6) as in (5.8) and take expectations. We then get that

$$\mathbb{E}[\text{OPT}(G)] \leq \sum_{b \in B, t \in [n]} \sum_{\boldsymbol{e} \in \mathcal{I}} \text{val}(\boldsymbol{e}) \cdot x_t(\boldsymbol{e} \,||\, b).$$

Now, $\text{OPT}(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n) = \mathbb{E}[\text{OPT}(G)]$ by definition, so since $(x_t(\boldsymbol{e} \,||\, b))_{b \in B, t \in [n], \boldsymbol{e} \in \mathcal{C}_b}$ is feasible, it holds that $\text{OPT}(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n) \leq \text{LPOPT}(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$, thus completing the proof. □

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

$$\widetilde{x}_{u,i}(b) := \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_b: \\ (u,b) \in \boldsymbol{e}}} q(\boldsymbol{e}_{<(u,b)}) \cdot x_i(\boldsymbol{e} \,||\, b). \tag{5.9}$$

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 (5.2) 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, $\frac{\sum_{\boldsymbol{e} \in \mathcal{C}_{v_i}} x_i(\boldsymbol{e} \,||\, v_i)}{r_i(v_i)} = 1$.

Using the above observation, we can 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 $\texttt{VertexProbe}(v_i, \partial(v_i), (x_i(\boldsymbol{e} \,||\, v_i)/r_i(v_i))_{\boldsymbol{e} \in \mathcal{C}_{v_i}})$. Observe then the following extension of Lemma 3.2:

**Lemma 5.2.** *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* $u \in U$,

$$\mathbb{P}[\texttt{VertexProbe}(v_i, \partial(v_i), (x_i(\boldsymbol{e} \,||\, v_i)/r_i(v_i))_{\boldsymbol{e} \in \mathcal{C}_{v_i}}) = (u, b) \mid v_i = b] = \frac{p_{u,b} \cdot \widetilde{x}_{u,i}(b)}{r_i(b)}.$$

**Definition 2** (Propose - Known I.D Instance)**.** We say that `VertexProbe` **proposes** $v_i$ to vertex $u \in \partial(v_i)$, provided the algorithm outputs $(u, v_i)$ when executing on online vertex $v_i$ for $i \in [n]$. When it is clear that `VertexProbe` is being executed on $v_i$, we say that $v_i$ proposes to $u$.

Consider now the generalization of Algorithm 1 where $\pi$ is generated either u.a.r. or adversarially.

**Algorithm 4** Known I.D

**Input:** a known i.d. input $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$.
**Output:** a matching $\mathcal{M}$ of active edges of $G \sim (H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$.

```
1: $\mathcal{M} \leftarrow \emptyset$.
2: Compute an optimum solution of LP-config-id for $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$, say $(x_i(\boldsymbol{e} \,||\, b))_{i \in [n], b \in B, \boldsymbol{e} \in \mathcal{C}_b}$.
3: for $t = 1, \ldots, n$ do
4:     Let $a \in B$ be the type of the current arrival $v_{\pi(t)}$.        ▷ to simplify notation
5:     Set $e \leftarrow \texttt{VertexProbe}\Big(v_{\pi(t)}, \partial(v_{\pi(t)}), \Big(x_{\pi(t)}(\boldsymbol{e} \,||\, a) \cdot r^{-1}_{\pi(t)}(a)\Big)_{\boldsymbol{e} \in \mathcal{C}_a}\Big)$.
6:     if $e = (u, v_{\pi(t)})$ for some $u \in U$, and $u$ is unmatched then
7:         Add $e$ to $\mathcal{M}$.
8: return $\mathcal{M}$.
```

Similarly, to Algorithm 1, one can show that Algorithm 4 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 a known i.i.d. instance – then it is relatively easy to show that this algorithm's competitive ratio improves to $1 - 1/e$.

**Proposition 5.3.** *If Algorithm 4 is presented a known i.i.d. input, say the type graph* $H_{typ}$ *together with the distribution* $\mathcal{D}$, *then* $\mathbb{E}[w(\mathcal{M})] \geq (1 - 1/e)\, OPT(H_{typ}, \mathcal{D})$.

For the case of non-identical distributions, we require online contention resolution schemes. Given an arbitrary integer $k \geq 1$, define the ground set $[k] := \{1, \ldots, k\}$. Fix $\boldsymbol{z} \in [0,1]^k$, and for each $i \in [k]$, let $Z_i$ be an indicator random variable which is 1 independently with probability

$z_i$. Let us denote $\mathcal{P} := \{\boldsymbol{z} \in [0,1]^k : \sum_{i=1}^k z_i \leq 1\}$. Note that $\mathcal{P}$ is the convex relaxation of the constraint imposed by the 1-uniform matroid on $[k]$.

**Definition 3** (Contention Resolution Scheme – 1-Uniform Matroid – [47])**.** A **contention resolution scheme** (CRS) for the 1-uniform matroid on $[k]$ is a (randomized) algorithm $\psi$, which given $\boldsymbol{z} \in \mathcal{P}$ and $S \subseteq [k]$ as inputs, returns at most one element $\psi(\boldsymbol{z}, S)$ of $S$. Given $\alpha \in [0,1]$, $\psi$ is said to be $\alpha$-**selectable**, provided for all $i \in [k]$ and $\boldsymbol{z} \in \mathcal{P}$,

$$\mathbb{P}[i = \psi(\boldsymbol{z}, R(\boldsymbol{z})) \mid i \in R(\boldsymbol{z})] \geq \alpha, \tag{5.10}$$

where the probability is over the generation of $R(\boldsymbol{z}) := \{j \in [k] : Z_j = 1\}$, and the potential randomness used by $\psi$. We say that $\psi$ is an **online** CRS, provided it is revealed $(Z_i)_{i=1}^k$ one-by-one in an (unknown) adversarially chosen order $\pi$. Upon learning $Z_{\pi(t)}$, it makes a irrevocable decision whether or not to return $\pi(t)$ as its output. A random CRS is an OCRS where the arrival order $\pi$ is instead drawn u.a.r. and independently of all other randomization.

During the execution of Algorithm 4, let $Z_{u,i}$ be the indicator random variable for the event in which $v_i$ proposes to vertex $u$. Then, $\mathbb{P}[Z_{u,i} = 1] = z_{u,i}$ where

$$z_{u,i} := \sum_{b \in B} p_{u,b} \cdot \widetilde{x}_{u,i}(b) = \sum_{b \in B} \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_b: \\ (u,b) \in \boldsymbol{e}}} p_{u,b} \cdot q(\boldsymbol{e}_{<(u,b)}) \cdot x_i(\boldsymbol{e} \,||\, b). \tag{5.11}$$

Moreover, for each fixed $u \in U$, the random variables $(Z_{u,i})_{i=1}^n$ are independent. We now prove that we can apply the same reduction to online (random order) contention resolution as in the known stochastic graph setting.

**Theorem 5.4.** *Given an $\alpha$-selectable OCRS (RCRS) for 1-uniform matroids, there exists an $\alpha$-competitive online probing algorithm for known i.d. instances and adversarial (random order) arrivals.*

*Proof of Theorem 5.4.* Let us first consider the case when $\pi$ is adversarially generated, and we are given an $\alpha$-selectable OCRS $\psi$. For notational simplicity, suppose that $\pi(t) = t$ for each $t \in [n]$, so that the online vertices arrive in order $v_1, \ldots, v_n$. We first describe the modification of Algorithm 4 using $|U|$ concurrent executions of $\psi$.

**Algorithm 5** Known I.D. – AOM – Modified

```
Input: a known i.d. input $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$.
Output: a matching $\mathcal{M}$ of active edges of $G \sim (H_{\text{typ}}, (\mathcal{D}_t)_{t=1}^n)$.
1: $\mathcal{M} \leftarrow \emptyset$.
2: Compute an optimum solution of LP-config-id for $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$, say $(x_i(\boldsymbol{e} \,||\, b))_{i \in [n], b \in B, \boldsymbol{e} \in \mathcal{C}_b}$.
3: for $t = 1, \ldots, n$ do
4:     Let $a \in B$ be the type of the current arrival $v_t$.
5:     Set $e \leftarrow$ VertexProbe$\left(v_t, \partial(v_t), \left(x_t(\boldsymbol{e} \,||\, a) \cdot r_t^{-1}(a)\right)_{\boldsymbol{e} \in \mathcal{C}_a}\right)$.
6:     if $e = (u, v_t)$ for some $u \in U$ then          ▷ $v_t$ proposes to $u$ (i.e., $Z_{u,t} = 1$)
7:         Execute $\psi$ on $(z_{u,v_i})_{i=1}^n$ and $(Z_{u,i})_{i=1}^t$, and add $e = (u, v_t)$ to $\mathcal{M}$ if $e$ is returned by $\psi$.
8: return $\mathcal{M}$.
```

We first verify that $\mathcal{M}$ is a matching of $G$. To see this, notice that $\psi$ returns at most once edge for each $u \in U$. Moreover, given $i \in [n]$, a necessary condition for $(u, v_i) \in \mathcal{M}$ is that `VertexProbe` must propose $v_i$ to $u$. However, for a fixed $v_i$, there is at most one vertex of $U$ for which this occurs.

To prove the algorithm is $\alpha$-competitive, first observe that 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, by Theorem 5.1, it suffices to show that

$$\mathbb{P}[(u, v_t) \in \mathcal{M} \text{ and } v_t = b] \geq \alpha \cdot p_{u,b} \widetilde{x}_{u,t}(b) \tag{5.12}$$

for each $u \in U, t \in [n]$ and $b \in B$. We may thus assume that $p_{u,b} \widetilde{x}_{u,t}(b) > 0$, where we note that $\mathbb{P}[v_t = b, Z_{u,t} = 1] = p_{u,b} \cdot \widetilde{x}_{u,t}(b)$.

Observe first that $(u, v_t) \in \mathcal{M}$ if and only if $(u, v_t)$ is returned by $\psi$ when it executes on $(z_{u,i})_{i=1}^n$ and $(Z_{u,i})_{i=1}^t$. Thus, since $\psi$ is an $\alpha$-selectable OCRS, by Definition 3,

$$\mathbb{P}[(u, v_t) \in \mathcal{M} \mid Z_{u,t} = 1] \geq \alpha. \tag{5.13}$$

We now show how this implies the stronger claim of (5.12). Observe that whether or not $\psi$ returns $(u, v_t)$ is a randomized function of $(z_{u,v})_{u \in U}$ and $(Z_{u,i})_{i=1}^t$. Crucially, $\psi$ does *not* make use of the type of $v_i$. As a result, conditional on $(Z_{u,i})_{i=1}^t$, the events $\{(u, v_t) \in \mathcal{M}\}$ and $\{v_t = b\}$ are independent, and so

$$\mathbb{P}[(u, v_t) \in \mathcal{M} \mid v_t = b, (Z_{u,i})_{i=1}^t] = \mathbb{P}[(u, v_t) \in \mathcal{M} \mid (Z_{u,i})_{i=1}^t]. \tag{5.14}$$

Now, taking expectations over $(Z_{u,i})_{i=1}^{t-1}$ in (5.14), we get that

$$\mathbb{P}[(u, v_t) \in \mathcal{M} \mid v_t = b, Z_{u,t} = 1] = \mathbb{P}[(u, v_t) \in \mathcal{M} \mid Z_{u,t} = 1]. \tag{5.15}$$

However, $\mathbb{P}[(u, v_t) \in \mathcal{M} \mid Z_{u,t} = 1] \geq \alpha$ by (5.13). Thus, $\mathbb{P}[(u, v_t) \in \mathcal{M} \mid v_t = b, Z_{u,t} = 1] \geq \alpha$, and so (5.12) holds by recalling that $\mathbb{P}[v_t = b, Z_{u,t} = 1] = p_{u,b} \cdot \widetilde{x}_{u,t}(b)$, and that $Z_{u,t} = 1$ is a necessary condition for $(u, v_t) \in \mathcal{M}$.

The reduction for random order arrivals proceeds identically, and so we omit the argument. □

For a 1-uniform matroid, Ezra et al. [23] prove the existence of a $1/2$-selectable OCRS, and Lee and Singla prove the existence of a $1 - 1/e$-selectable RCRS. Combining these results with Theorem 5.4, we get the claimed competitive ratios of Table 1.

**Corollary 5.5.** *In the AOM, there exists a $1/2$-competitive online probing algorithm for the online stochastic matching problem with known i.d. instances.*

**Corollary 5.6.** *In the ROM, there exists a $1 - 1/e$-competitive online probing algorithm for the online stochastic matching problem with known i.d. instances.*

## 6 Edge-weighted Worst-case Instance Model

Let us suppose that $G = (U, V, E)$ is an adversarially generated stochastic graph. Denote the online nodes of $V$ by $v_1, \ldots, v_n$, where the order is generated u.a.r, and define $V_t = \{v_1, \ldots, v_t\}$ to be the first $t$ arrivals of $V$. Moreover, set $G_t := G[U \cup V_t]$, and $\text{LPOPT}(G_t)$ as the value of an optimal solution to LP-config (this is a random variable, as $V_t$ is a random subset of $V$). The following inequality then holds:

**Lemma 6.1.** *For each $t \geq 1$, $\mathbb{E}[LPOPT(G_t)] \geq \frac{t}{n} \, LPOPT(G)$.*

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

**Algorithm 6** Edge-weighted Worst-case Instance ROM

```
Input: $U$ and $n := |V|$.
Output: a matching $\mathcal{M}$ from the (unknown) stochastic graph $G = (U, V, E)$ of active edges.
 1: Set $\mathcal{M} \leftarrow \emptyset$.
 2: Set $G_0 = (U, \emptyset, \emptyset)$
 3: for $t = 1, \ldots, n$ do
 4:     Input $v_t$, with $(w_e)_{e \in \partial(v_t)}$, $(p_e)_{e \in \partial(v_t)}$ and $\mathcal{C}_{v_t}$.
 5:     Compute $G_t$, by updating $G_{t-1}$ to contain $v_t$ (and its relevant information).
 6:     if $t < \lfloor n/e \rfloor$ then
 7:         Pass on $v_t$.
 8:     else
 9:         Solve LP-config for $G_t$ and find an optimal solution $(x_v(\boldsymbol{e}))_{v \in V_t, \boldsymbol{e} \in \mathcal{C}_v}$.
10:         Set $e_t \leftarrow \texttt{VertexProbe}(v_t, \partial(v_t), (x_v(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_{v_t}})$.
11:         if $e_t = (u_t, v_t) \neq \emptyset$ and $u_t$ is unmatched then
12:             Add $e_t$ to $\mathcal{M}$.
13: return $\mathcal{M}$.
```

**Remark 6.2.** Unlike the algorithm of Kesselheim et al., our algorithm is randomized, and the polytope LP-config does not always have an optimal integral solution. We leave it as an interesting open question as to whether or not Algorithm 6 can be derandomized.

**Theorem 6.3.** *If $\mathcal{M}$ is the matching returned by Algorithm 6 when executing on $G$, then*

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

*provided the vertices of $V$ arrive u.a.r.*

Let us consider the matching $\mathcal{M}$ returned by the algorithm, as well as its weight, which we recall is denoted $w(\mathcal{M})$. Set $\alpha := 1/e$ for clarity, and take $t \geq \lceil \alpha n \rceil$. For each $\alpha n \leq t \leq n$, define $R_t$ as the *unmatched vertices* of $U$ when vertex $v_t$ arrives. Note that proposing $v_t$ to $u_t$ is necessary, but not sufficient, for $v_t$ to match to $u_t$. Define $w(e_t) := w_{e_t} \cdot \mathbf{1}_{[e_t \neq \emptyset]}$. With this notation, we have that $\mathbb{E}[w(\mathcal{M})] = \sum_{t=\alpha n}^{n} \mathbb{E}[w(u_t, v_t) \cdot \mathbf{1}_{[u_t \in R_t]}]$. Moreover, we claim the following:

**Lemma 6.4.** *For each $t \geq \lceil \alpha n \rceil$, $\mathbb{E}[w(e_t)] \geq LPOPT(G)/n$.*

**Lemma 6.5.** *For each $t \geq \lceil \alpha n \rceil$, define $f(t, n) := \lfloor \alpha n \rfloor / (t-1)$. In this case, $\mathbb{P}[u_t \in R_t \,|\, V_t, v_t] \geq f(t, n)$, where $V_t = \{v_1, \ldots, v_t\}$ and $v_t$ is the $t^{th}$ arriving node of $V$* [5].

The proofs of Lemmas 6.4 and 6.5 mostly follow the analogous claims as proven by Kesselheim et al. in the secretary matching problem. We present formal proofs in the Appendix B. Assuming these lemmas, the proof of Theorem 6.3 follows easily:

*Proof of Theorem 6.3.* Let us consider the matching $\mathcal{M}$ returned by the algorithm, as well as its weight, which we denote by $w(\mathcal{M})$. Set $\alpha := 1/e$ for clarity, and take $t \geq \lceil \alpha n \rceil$, where we define $R_t$ to be the *unmatched vertices* of $U$ when vertex $v_t$ arrives. Moreover, define $e_t := (u_t, v_t)$, where

[5] Note that since $V_t$ is a set, conditioning on $V_t$ only reveals which vertices of $V$ encompass the first $t$ arrivals, *not* the order they arrived in. Hence, conditioning on $v_t$ as well reveals strictly more information.

$u_t$ is the vertex of $U$ which $v_t$ proposes to, which is the empty-set by definition if no such proposal is made. Observe that

$$\mathbb{E}[w(\mathcal{M})] = \sum_{t=\lceil \alpha n \rceil}^{n} \mathbb{E}[w(u_t, v_t) \cdot \mathbf{1}_{[u_t \in R_t]}]. \tag{6.1}$$

Fix $\lceil \alpha n \rceil \leq t \leq n$, and first observe that $w(u_t, v_t)$ and $\{u_t \in R_t\}$ are conditionally independent given $(V_t, v_t)$, as the probes involving $\partial(v_t)$ are independent from those of $v_1, \ldots, v_{t-1}$. Thus,

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

Moreover, Lemma 6.5 implies that $\mathbb{E}[w(u_t, v_t) \,|\, V_t, v_t] \cdot \mathbb{P}[u_t \in R_t \,|\, V_t, v_t] \geq \mathbb{E}[w(u_t, v_t) \,|\, V_t, v_t] f(t, n)$, and so $\mathbb{E}[w(u_t, v_t)\, \mathbf{1}_{[u_t \in R_t]} \,|\, V_t, v_t] \geq \mathbb{E}[w(u_t, v_t) \,|\, V_t, v_t]\, f(t, n)$. Thus, by the law of iterated expectations[6]

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

As a result, using (6.1), $\mathbb{E}[w(\mathcal{M})] = \sum_{t=\lceil \alpha n \rceil}^{n} \mathbb{E}[w(u_t, v_t)\, \mathbf{1}_{[u_t \in R_t]}] \geq \sum_{t=\lceil \alpha n \rceil}^{n} f(t, n)\, \mathbb{E}[w(u_t, v_t)]$. We may thus conclude that

$$\mathbb{E}[w(\mathcal{M})] \geq \text{LPOPT}(G) \sum_{t=\lceil \alpha n \rceil}^{n} \frac{f(t, n)}{n},$$

after applying Lemma 6.4. As $\sum_{t=\lceil \alpha n \rceil}^{n} f(t, n)/n \geq (1/e - 1/n)$, the result holds. □

## 7 Vertex-weighted Worst-case Instance Model

In this section, we analyze a *greedy* online probing algorithm when the stochastic graph $G = (U, V, E)$ has (offline) vertex weights $(w_u)_{u \in U}$. Upon the arrival of $s$, the probes to $\partial(s)$ are made in such a way that $s$ gains as large a match as possible (in expectation), provided the currently unmatched nodes of $U$ are equal to $R \subseteq U$. As such, we must follow the probing strategy of OPT when restricted to the **induced stochastic graph**[7] $G[\{s\} \cup R]$. We denote the performance of OPT on $G[\{s\} \cup R]$ by $\text{OPT}(R, s)$.

Assume that $R = U$, and that $w_{u,s} := w_u$ for each $u \in U$ such that $(u, s) \in \partial(s)$. Recall that for $\boldsymbol{e} \in \mathcal{C}_s$,

$$\text{val}(\boldsymbol{e}) := \sum_{i=1}^{|\boldsymbol{e}|} p_{e_i} w_{e_i} \prod_{j=1}^{i-1} (1 - p_{e_i}). \tag{7.1}$$

Observe that if one probes the edges of $\boldsymbol{e}$ in order, then $\text{val}(\boldsymbol{e})$ is the expected weight of the first active edge of $\boldsymbol{e}$. This is a **non-adaptive** probing algorithm for $G[U \cup \{s\}]$. Since OPT operates in the probe-commit model, there exists a non-adaptive probing algorithm with (optimal) performance $\text{OPT}(s, U)$. Moreover, we show such a strategy can be found efficiently assuming that $\mathcal{C}_s$ is downward-closed.

[6] $\mathbb{E}[w(u_t, v_t) \cdot \mathbf{1}_{[u_t \in R_t]} \,|\, V_t, v_t]$ is a random variable which depends on $V_t$ and $v_t$, and so the outer expectation is over the randomness in $V_t$ and $v_t$.

[7] Given $R \subseteq U, V' \subseteq V$, the induced stochastic graph $G[R \cup V']$ is formed by restricting the edges weights and probabilities of $G$ to those edges within $R \times V'$. Similarly, each probing constraint $\mathcal{C}_s$ is restricted to those strings whose entries lie entirely in $R \times \{s\}$.

**Theorem 7.1.** *There exists a dynamic programming (DP) based algorithm DP-OPT, which given access to $G[\{s\} \cup U]$, computes a tuple $\mathbf{e}' \in \mathcal{C}_s$, such that $OPT(s, U) = val(\mathbf{e}')$. Moreover, DP-OPT executes in time $O(|U|^2)$, assuming access to a membership oracle for the downward-closed constraint $\mathcal{C}_s$.*

*Proof.* Our goal is to show that $\boldsymbol{e}'$ can be computed efficiently. Now, for any $\boldsymbol{e} \in \mathcal{C}_s$, let $\boldsymbol{e}^r$ be the rearrangement of $\boldsymbol{e}$, based on the non-increasing order of the weights $(w_e)_{e \in \boldsymbol{e}}$. Since $\mathcal{C}_s$ is downward-closed, we know that $\boldsymbol{e}^r$ is also in $\mathcal{C}_s$. Moreover, $\text{val}(\boldsymbol{e}^r) \geq \text{val}(\boldsymbol{e})$ (following observations in [44, 14]). Hence, let us order the edges of $\partial(s)$ as $e_1, \ldots, e_m$, such that $w_{e_1} \geq \ldots \geq w_{e_m}$, where $m := |\partial(s)|$. Observe then that it suffices to maximize (7.1) over those strings within $\mathcal{C}_s$ which respect this ordering on $\partial(s)$. Stated differently, let us denote $\mathcal{I}_s$ as the family of subsets of $\partial(s)$ induced by $\mathcal{C}_s$, and define the set function $f : 2^{\partial(s)} \rightarrow [0, \infty)$, where $f(B) := \text{val}(\boldsymbol{b})$ for $B = \{b_1, \ldots, b_{|B|}\} \subseteq \partial(s)$, such that $\boldsymbol{b} = (b_1, \ldots, b_{|B|})$ and $w_{b_1} \geq \ldots \geq w_{b_{|B|}}$. Our goal is then to efficiently maximize $f$ over the set-system $(\partial(s), \mathcal{I}_s)$. Observe that $\mathcal{I}_s$ is downward-closed and that we can simulate oracle access to $\mathcal{I}_s$, based on our oracle access to $\mathcal{C}_s$.

For each $i = 0, \ldots, m-1$, denote $\partial(s)^{>i} := \{e_{i+1}, \ldots, e_m\}$, and $\partial(s)^{>m} := \emptyset$. Moreover, define the family of subsets $\mathcal{I}_s^{>i} := \{B \subseteq \partial(s)^{>i} : B \cup \{e_i\} \in \mathcal{I}_s\}$ for each $1 \leq i \leq m$, and $\mathcal{I}_s^{>0} := \mathcal{I}_s$. Observe then that $(\partial(s)^{>i}, \mathcal{I}_s^{>i})$ is a downward-closed set system, as $\mathcal{I}_s$ is downward-closed. Moreover, we may simulate oracle access to $\mathcal{I}_s^{>i}$ based on our oracle access to $\mathcal{I}_s$.

Denote $\text{OPT}(\mathcal{I}_s^{>i})$ as the maximum value of $f$ over constraints $\mathcal{I}_s^{>i}$. Observe then that for each $0 \leq i \leq m-1$, the following recursion holds:

$$\text{OPT}(\mathcal{I}_s^{>i}) := \max_{j \in \{i+1, \ldots, m\}} (p_{e_j} \cdot w_{e_j} + (1 - p_{e_j}) \cdot \text{OPT}(\mathcal{I}_s^{>j})) \tag{7.2}$$

Hence, given access to the values $\text{OPT}(\mathcal{I}_s^{>i+1}), \ldots, \text{OPT}(\mathcal{I}_s^{>m})$, we can compute $\text{OPT}(\mathcal{I}_s^{>i})$ efficiently. Moreover, $\text{OPT}(\mathcal{I}_s^{>m}) = 0$ by definition. Thus, it is clear that we can use (7.2) to recover an optimal solution to $f$. We can define DP-OPT to be a memoization based implementation of (7.2). It is clear DP-OPT can be implemented in the claimed time complexity. □

Given $R \subseteq U$, denote the output of executing DP-OPT on $G[\{s\} \cup R]$ by DP-OPT$(s, R)$. Consider now the following online probing algorithm:

**Algorithm 7** Greedy-DP

```
Input: offline vertices $U$ with vertex weights $(w_u)_{u \in U}$.
Output: a matching $\mathcal{M}$ of active edges of the unknown stochastic graph $G = (U, V, E)$.
 1: $\mathcal{M} \leftarrow \emptyset$.
 2: $R \leftarrow U$.
 3: for $t = 1, \ldots, n$ do
 4:     Let $v_t$ be the current online arrival node, with constraint $\mathcal{C}_{v_t}$.
 5:     Set $\boldsymbol{e} \leftarrow$ DP-OPT$(v_t, R)$
 6:     for $i = 1, \ldots, |\boldsymbol{e}|$ do
 7:         Probe $e_i$.
 8:         if st$(e_i) = 1$ then
 9:             Add $e_i$ to $\mathcal{M}$, and update $R \leftarrow R \setminus \{u_i\}$, where $e_i = (u_i, v_t)$.
10: return $\mathcal{M}$.
```

**Theorem 7.2.** *If $\mathcal{M}$ is the matching returned by Algorithm 7 when executing on vertex-weighted stochastic graph $G$, then*

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

*provided the vertices of $V$ arrive in adversarial order.*

In order to analyze Algorithm 7, we first upper bound $\text{OPT}(G)$ using an LP relaxation which extends the LP from [14] to online probing constraints. For each $u \in U$ and $v \in V$, let $x_{u,v}$ be a decision variable corresponding to the probability that OPT probes the edge $(u, v)$.

$$\begin{aligned}
\text{maximize} \quad & \sum_{u \in U} \sum_{v \in V} w_u \cdot p_{u,v} \cdot x_{u,v} && && \text{(LP-DP)} \\
\text{subject to} \quad & \sum_{v \in V} p_{u,v} \cdot x_{u,v} \leq 1 && \forall u \in U && (7.3) \\
& \sum_{u \in R} w_u \cdot p_{u,v} \cdot x_{u,v} \leq \text{OPT}(v, R) && \forall v \in V,\, R \subseteq U && (7.4) \\
& x_{u,v} \geq 0 && \forall u \in U, v \in V && (7.5)
\end{aligned}$$

Denote $\text{LPOPT}_{\text{DP}}(G)$ as the optimal value of this LP. Constraint (7.3) can be viewed as ensuring that the expected number of matches made to $u \in U$ is at most 1. Similarly, (7.4) can be interpreted as ensuring that expected stochastic reward of $v$, suggested by the solution $(x_{u,v})_{u \in U, v \in V}$, is actually attainable by the adaptive benchmark. More precisely, given $R \subseteq U$ and $v \in V$, if one restricts the match of $v$ made by OPT to $R \times \{v\}$, then its expected value cannot exceed $\text{OPT}(v, R)$. This implies the following (see [14] for a detailed proof specific to patience values):

**Lemma 7.3.** *$OPT(G) \leq LPOPT_{DP}(G)$*

In [14], the performance of Algorithm 7 is compared to a solution to LP-DP to prove the algorithm is 1/2-competitive for patience values. It is easily verified that this argument extends to online probing constraints. We instead prove Theorem 7.2 using a primal-dual charging argument based on the dual of LP-DP, as this allows us to introduce the techniques needed for the analysis of Algorithm 7 in the ROM.

For each $u \in U$, define the variable $\alpha_u$. Moreover, for each $R \subseteq U$ and $v \in V$, define the variable $\phi_{v,R}$ (these latter variables correspond to constraint (7.4)):

$$\begin{aligned}
\text{minimize} \quad & \sum_{u \in U} \alpha_u + \sum_{v \in V} \sum_{R \subseteq U} \text{OPT}(v, R) \cdot \phi_{v,R} && && \text{(LP-dual-DP)} \\
\text{subject to} \quad & p_{u,v} \cdot \alpha_u + \sum_{\substack{R \subseteq U: \\ u \in R}} w_u \cdot p_{u,v} \cdot \phi_{v,R} \geq w_u \cdot p_{u,v} && \forall u \in U, v \in V && (7.6) \\
& \alpha_u \geq 0 && \forall u \in U && (7.7) \\
& \phi_{v,R} \geq 0 && \forall v \in V, R \subseteq U && (7.8)
\end{aligned}$$

Let $((\alpha_u)_{u \in U}, (\phi_{v,R})_{v \in V, R \subseteq U})$ be a dual solution which is initially identically 0. As Algorithm 7 executes, we modify it in the following way. Fix $v \in V, u \in U$, and $R \subseteq U$, where $u \in R$. If $R$ consists of the unmatched vertices when $v$ arrives, then suppose that Algorithm 7 matches $v$ to $u$ while making its probes to a subset of the edges of $R \times \{v\}$. In this case, we **charge** $w_u$ to $\alpha_u$ and

$w_u/\text{OPT}(v,R)$ to $\phi_{v,R}$. Observe that each subset $R \subseteq U$ is charged at most once, as is each $u \in U$. Moreover,

$$\mathbb{E}[w(\mathcal{M})] = \frac{1}{2} \cdot \left( \sum_{u \in U} \mathbb{E}[\alpha_u] + \sum_{v \in V} \sum_{R \subseteq U} \text{OPT}(v,R) \cdot \mathbb{E}[\phi_{v,R}] \right), \tag{7.9}$$

where the expectation is over $(\text{st}(e))_{e \in E}$. Let us now set $\alpha_u^* := \mathbb{E}[\alpha_u]$ and $\phi_{v,R}^* := \mathbb{E}[\phi_{v,R}]$ for $u \in U, v \in V$ and $R \subseteq U$. We prove the following in Appendix B:

**Lemma 7.4.** *$((\alpha_u^*)_{u \in U}, (\phi_{v,R}^*)_{v \in V, R \subseteq U})$ is a feasible solution to LP-dual-DP.*

Theorem 7.2 then follows immediately from (7.9), Lemma 7.3 and Lemma 7.4.

## 7.1 Analyzing Algorithm 7 in the ROM

In general, the behaviour of OPT on $G[\{s\} \cup R]$ can change very much, even for minor changes to $R$. For instance, if $R = U$, then OPT may probe the edge $(u,s)$ first – thus giving it highest priority – whereas if $u^* \in U$ is removed from $U$ (where $u^* \neq u$), OPT may not probe $(u,v)$ at all. See the below example for an explicit instance of this behaviour.

**Example 7.5.** *Let $G = (U,V,E)$ be a bipartite graph with $U = \{u_1,u_2,u_3,u_4\}$, $V = \{v\}$ and $\ell_v = 2$. Set $p_{u_1,v} = 1/3$, $p_{u_2,v} = 1$, $p_{u_3,v} = 1/2$, $p_{u_4,v} = 2/3$. Fix $\varepsilon > 0$, and let the weights of offline vertices be $w_{u_1} = 1+\varepsilon$, $w_{u_2} = 1+\varepsilon/2$, $w_{u_3} = w_{u_4} = 1$. We assume that $\varepsilon$ is sufficiently small – concretely, $\varepsilon \leq 1/12$. If $R_1 := U$, then OPT probes $(u_1,v)$ and then $(u_2,v)$ in order. On the other hand, if $R_2 = R_1 \setminus \{u_2\}$, then OPT does not probe $(u_1,v)$. Specifically, OPT probes $(u_3,v)$ and then $(u_4,v)$.*

Using Example 7.5, it is easy to consider an execution of Algorithm 7 on $G$ where $v$ is matched to $u$, but if a new vertex $v^*$ is added to $G$ ahead of $v$, $u$ is never matched. We thus refer to Algorithm 7 as being **non-monotonic**. This contrasts with the classical setting, in which the deterministic greedy algorithm in the ROM setting does not exhibit this behaviour, and thus is **monotonic**. The absence of monotonicity isn't problematic in the adversarial setting of Theorem 7.2 because our charging argument does not depend on the order of the online vertex arrivals. On the other hand, in the ROM we construct a solution to LP-dual-DP using a function $g : [0,1] \to [0,1]$ which depends on the arrival time of each online node. In order for this solution to be feasible in expectation, we (provably) must restrict to collections of stochastic graphs in which executions of Algorithm 7 are monotonic. This leads us to the definition of *rankability*, which characterizes a large number of such graphs.

Given a vertex $v \in V$, and an ordering $\pi_v$ on $\partial(v)$, if $R \subseteq U$, then define $\pi_v(R)$ to be the longest string constructible by iteratively appending the edges of $R \times \{v\}$ via $\pi_v$, subject to respecting constraint $\mathcal{C}_v^R$. More precisely, given $\boldsymbol{e}'$ after processing $e_1, \ldots, e_i$ of $R \times \{v\}$ ordered according to $\pi_v$, if $(\boldsymbol{e}', e_{i+1}) \in \mathcal{C}_v^R$, then update $\boldsymbol{e}'$ by appending $e_{i+1}$ to its end, otherwise move to the next edge $e_{i+2}$ in the ordering $\pi_v$, assuming $i+2 \leq |R|$. If $i+2 > |R|$, return the current string $\boldsymbol{e}'$ as $\pi_v(R)$. We say that $v$ is **rankable**, provided there exists a choice of $\pi_v$ which depends *solely* on $(p_e)_{e \in \partial(v)}$, $(w_e)_{e \in \partial(v)}$ and $\mathcal{C}_v$, such that for *every* $R \subseteq U$, the strings DP-OPT$(v,R)$ and $\pi_v(R)$ are equal. Crucially, if $v$ is rankable, then when vertex $v$ arrives while executing Algorithm 7, one can compute the ranking $\pi_v$ on $\partial(v)$ and probe the adjacent edges of $R \times \{v\}$ based on this order, subject to not violating the constraint $\mathcal{C}_v^R$. By following this probing strategy, the optimality of DP-OPT ensures that the expected weight of the match made to $v$ will be OPT$(v,R)$. We consider three (non-exhaustive) examples of rankability:

**Proposition 7.6.** *Let $G = (U, V, E)$ be a stochastic graph, and suppose that $v \in V$. If $v$ satisfies either of the following conditions, then $v$ is rankable:*

1. *$v$ has unit patience or unlimited patience; that is, $\ell_v \in \{1, |U|\}$.*
2. *$v$ has patience $\ell_v$, and for each $u_1, u_2 \in U$, if $p_{u_1,v} \leq p_{u_2,v}$ then $w_{u_1} \leq w_{u_2}$.*
3. *$G$ is unweighted, and $v$ has a budget $B_v$ with edge probing costs $(b_{u,v})_{u \in U}$, and for each $u_1, u_2 \in U$, if $p_{u_1,v} \leq p_{u_2,v}$ then $b_{u_1,v} \geq b_{u_2,v}$.*

The rankable assumption is similar to assumptions referred to as laminar, agreeable and compatible in other applications. We refer to the stochastic graph $G$ as **rankable**, provided all of its online vertices are themselves rankable. We emphasize that distinct vertices of $V$ may each use their own separate rankings of their adjacent edges. Our next result proves a $1 - 1/e$ competitive ratio $G$ is rankable, as well as an asymptotic competitive ratio of $1 - 1/e$ provided the edge probabilities of $G$ tend to 0 sufficiently fast (as $|G| \to \infty$).

**Theorem 7.7.** *Suppose Algorithm 7 returns the matching $\mathcal{M}$ when executed on the vertex-weighted stochastic graph $G = (U, V, E)$ with random order vertex arrivals.*

1. *If $G$ is rankable, then*
$$\mathbb{E}[w(\mathcal{M})] \geq \left(1 - \frac{1}{e}\right) \cdot OPT(G).$$
2. *If $c_v := \max_{\boldsymbol{e} \in \mathcal{C}_v} |\boldsymbol{e}|$ and $p_v := \max_{e \in \partial(v)} p_e$, then*
$$\mathbb{E}[w(\mathcal{M})] \geq \min_{v \in V} (1 - p_v)^{c_v} \cdot \left(1 - \frac{1}{e}\right) \cdot OPT(G).$$
*Thus, if $\max_{v \in V} c_v \cdot p_v \to 0$, then $\mathbb{E}[w(\mathcal{M})] \geq (1 - o(1))\,(1 - 1/e) \cdot OPT(G)$.*

**Remark 7.8.** The analysis of Algorithm 7 is tight, as an execution of Algorithm 7 corresponds to the seminal Karp et al. [34] Ranking algorithm for unweighted non-stochastic (i.e., $p_e \in \{0, 1\}$ for all $e \in E$) bipartite matching.

**Corollary 7.9.** *Suppose $G = (U, V, E)$ is a vertex-weighted stochastic graph with unit patience values. If $\mathcal{M}$ is the matching returned by Algorithm 7 when executing on $G$, then*
$$\mathbb{E}[w(\mathcal{M})] \geq \left(1 - \frac{1}{e}\right) OPT(G),$$
*provided the vertices of $V$ arrive in random order.*

**Remark 7.10.** The guarantee of Theorem 7.7 is proven against $\text{LPOPT}_{\text{DP}}(G)$. In the special case when $G$ has unit patience, $\text{LPOPT}_{\text{std}}(G) \leq \text{LPOPT}_{\text{DP}}(G)$, where $\text{LPOPT}_{\text{std}}(G)$ is the value of an optimal solution to LP-std-unit on $G$. Thus, Corollary 7.9 implies that the $0.621 < 1 - 1/e$ in-approximation of Mehta and Panigraphi [41] against LP-std-unit does not apply to the ROM setting.

## 7.2 Proving Theorem 7.7

The dual-fitting argument used to prove Theorem 7.7 has an initial set-up which based on the argument in Devanur et al. [21]. First define $g : [0,1] \to [0,1]$ where $g(z) := \exp(z-1)$ for $z \in [0,1]$. We shall use $g$ to perform our charging. Moreover, recall that given $v \in V$, we defined $c_v := \max_{\boldsymbol{e} \in \mathcal{C}_v} |\boldsymbol{e}|$ and $p_v := \max_{e \in \partial(v)} p_e$. Using these definitions, we define our target competitive ratio $F = F(G)$, where

$$F(G) := \begin{cases} 1 - \frac{1}{e} & \text{if } G \text{ is rankable,} \\ \left(1 - \frac{1}{e}\right) \cdot \min_{v \in V} (1 - p_v)^{c_v} & \text{otherwise.} \end{cases} \tag{7.10}$$

In order to prove Theorem 7.7, we shall prove that Algorithm 7 returns a matching of expected weight at least $F(G) \cdot \text{LPOPT}_{\text{DP}}(G)$ when executing on the stochastic graph $G$ in the ROM. Clearly, we may assume $F(G) > 0$, as otherwise there is nothing to prove, so we shall make this assumption for the rest of the section. Note that $F(G) \leq 1 - 1/e$ no matter the stochastic graph $G$.

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 Algorithm 7 in a non-decreasing order, based on the values of $(Y_v)_{v \in V}$. We now describe how the charging assignments are made while Algorithm 7 executes on $G$. First, we initialize a dual solution $((\alpha_u)_{u \in U}, (\phi_{v,R})_{v \in V, R \subseteq U})$ where all the variables are set equal to 0. Next, we take $v \in V, u \in U$, and $R \subseteq U$, where $u \in R$. If $R$ consists of the unmatched vertices of $v$ when it arrives at time $Y_v$, then suppose that Algorithm 7 matches $v$ to $u$ while making its probes to a subset of the edges of $R \times \{v\}$. In this case, we **charge** $w_u \cdot (1 - g(Y_v))/F$ to $\alpha_u$ and $w_u \cdot g(Y_v)/(F \cdot \text{OPT}(v,R))$ to $\phi_{v,R}$. Observe that each subset $R \subseteq U$ is charged at most once, as is each $u \in U$. Thus,

$$\mathbb{E}[w(\mathcal{M})] = F \cdot \left( \sum_{u \in U} \mathbb{E}[\alpha_u] + \sum_{v \in V} \sum_{R \subseteq U} \text{OPT}(v,R) \cdot \mathbb{E}[\phi_{v,R}] \right), \tag{7.11}$$

where the expectation is over the random variables $(Y_v)_{v \in V}$ and $(\text{st}(e))_{e \in E}$. If we now set $\alpha_u^* := \mathbb{E}[\alpha_u]$ and $\phi_{v,R}^* := \mathbb{E}[\phi_{v,R}]$ for $u \in U, v \in V$ and $R \subseteq U$, then (7.11) implies the following lemma:

**Lemma 7.11.** *Suppose $G = (U,V,E)$ is a stochastic graph for which Algorithm 7 returns the matching $\mathcal{M}$ when presented $V$ based on $(Y_v)_{v \in V}$ generated u.a.r. from $[0,1]$. In this case, if the variables $((\alpha_u^*)_{u \in U}, (\phi_{v,R}^*)_{v \in V, R \subseteq U})$ are defined through the above charging scheme, then*

$$\mathbb{E}[w(\mathcal{M})] = F \cdot \left( \sum_{u \in U} \alpha_u^* + \sum_{v \in V} \sum_{R \subseteq U} OPT(v,R) \cdot \phi_{v,R}^* \right).$$

We claim the following regarding $((\alpha_u^*)_{u \in U}, (\phi_{v,R}^*)_{v \in V, R \subseteq U})$:

**Lemma 7.12.** *If the online nodes of $G = (U,V,E)$ are presented to Algorithm 7 based on $(Y_v)_{v \in V}$ generated u.a.r. from $[0,1]$, then the solution $((\alpha_u^*)_{u \in U}, (\phi_{v,R}^*)_{v \in V, R \subseteq U})$ is a feasible solution to LP-dual-DP.*

Since LP-DP is a relaxation of the adaptive benchmark, Theorem 7.7 follows from Lemmas 7.11 and 7.12 in conjunction with weak duality.

### 7.2.1 Proving Dual Feasibility: Lemma 7.12

Let us suppose that the variables $((\alpha_u)_{u \in U}, (\phi_{v,R})_{v \in V, R \subseteq U})$ are defined as in the charging scheme of Section 7.2. In order to prove Lemma 7.12, we must show that for each fixed $u_0 \in U$ and $v_0 \in V$, we have that

$$\mathbb{E}[p_{u_0,v_0} \cdot \alpha_{u_0} + w_{u_0} \cdot p_{u_0,v_0} \sum_{\substack{R \subseteq U: \\ u_0 \in R}} \phi_{v_0,R}] \geq w_{u_0} \cdot p_{u_0,v_0}. \tag{7.12}$$

Our strategy for proving (7.12) first involves the same approach as used in Devanur et al. [21]. Specifically, we define the stochastic graph $\widetilde{G} := (U, \widetilde{V}, \widetilde{E})$, where $\widetilde{V} := V \setminus \{v_0\}$ and $\widetilde{G} := G[U \cup \widetilde{V}]$. We wish to compare the execution of the algorithm on the instance $\widetilde{G}$ to its execution on the instance $G$. It will be convenient to couple the randomness between these two executions by making the following assumptions:

1. For each $e \in \widetilde{E}$, $e$ is active in $\widetilde{G}$ if and only if it is active in $G$.
2. The same random variables, $(Y_v)_{v \in \widetilde{V}}$, are used in both executions.

If we now focus on the execution of $\widetilde{G}$, then define the random variable $\widetilde{Y}_c$ where $\widetilde{Y}_c := Y_{v_c}$ if $u_0$ is matched to some $v_c \in \widetilde{V}$, and $\widetilde{Y}_c := 1$ if $u_0$ remains unmatched after the execution on $\widetilde{G}$. We refer to the random variable $\widetilde{Y}_c$ as the **critical time** of vertex $u_0$ with respect to $v_0$. We claim the following lower bounds on $\alpha_{u_0}$ in terms of the critical time $\widetilde{Y}_c$.

**Proposition 7.13.**

- *If $G$ is rankable, then* $\alpha_{u_0} \geq \left(1 - \frac{1}{e}\right)^{-1} w_{u_0}(1 - g(\widetilde{Y}_c))$.
- *Otherwise,* $\mathbb{E}[\alpha_{u_0} \mid (Y_v)_{v \in V}, (st(e))_{e \in \widetilde{E}}] \geq \left(1 - \frac{1}{e}\right)^{-1} w_{u_0}(1 - g(\widetilde{Y}_c))$.

**Remark 7.14.** Note that the proof of Proposition 7.13 is the only part of the proof of Theorem 7.7 which depends on whether or not $G$ is rankable.

*Proof of Proposition 7.13.* For each $v \in V$, denote $R_v^{\mathrm{af}}(G)$ as the unmatched (remaining) vertices of $U$ right after $v$ is processed (attempts its probes) in the execution on $G$. We emphasize that if a probe of $v$ yields an active edge, thus matching $v$, then this match is excluded from $R_v^{\mathrm{af}}(G)$. Similarly, define $R_v^{\mathrm{af}}(\widetilde{G})$ in the same way for the execution on $\widetilde{G}$ (where $v$ is now restricted to $\widetilde{V}$).

We first consider the case when $G$ is rankable, and so $F(G) = 1 - 1/e$. Observe that since the constraints $(\mathcal{C}_v)_{v \in V}$ are substring-closed, we can use the coupling between the two executions to inductively prove that

$$R_v^{\mathrm{af}}(G) \subseteq R_v^{\mathrm{af}}(\widetilde{G}), \tag{7.13}$$

for each $v \in \widetilde{V}$ [8]. Now, since $g(1) = 1$ (by assumption), there is nothing to prove if $\widetilde{Y}_c = 1$. Thus, we may assume that $\widetilde{Y}_c < 1$, and as a consequence, that there exists some vertex $v_c \in V$ which matches to $u_0$ at time $\widetilde{Y}_c$ in the execution on $\widetilde{G}$.

On the other hand, by assumption we know that $u_0 \notin R_{v_c}^{\mathrm{af}}(\widetilde{G})$ and thus by (7.13), that $u_0 \notin R_{v_c}^{\mathrm{af}}(G)$. As such, there exists some $v' \in V$ which probes $(u_0, v')$ and succeeds in matching to $u_0$ at time $Y_{v'} \leq \widetilde{Y}_c$. Thus, since $g$ is monotone,

$$\alpha_{u_0} \geq \left(1 - \frac{1}{e}\right)^{-1} w_{u_0} \cdot (1 - g(Y_{v'})) \cdot \mathbf{1}_{[\widetilde{Y}_c < 1]} \geq \left(1 - \frac{1}{e}\right)^{-1} w_{u_0} \cdot (1 - g(\widetilde{Y}_c)),$$

[8] Example 7.5 shows why (7.13) will not hold if $G$ is not rankable.

and so the rankable case is complete.

We now consider the case when $G$ is not rankable. Suppose that $\mathcal{M}(v_0)$ is the vertex matched to $v_0$ when the algorithm executes on $G$, where $\mathcal{M}(v_0) := \emptyset$ provided no match is made. Observe then that if no match is made to $v_0$ in this execution, then the execution proceeds identically to the execution on $\widetilde{G}$. As a result, we get the following relation: $\alpha_{u_0} \geq \frac{w_{u_0}}{F}(1 - g(\widetilde{Y}_c)) \cdot \mathbf{1}_{[\mathcal{M}(v_0)=\emptyset]}$. Now, let us condition on $(\text{st}(e))_{e \in \widetilde{E}}$ and $(Y_v)_{v \in V}$, and recall the definitions of $p_{v_0} := \max_{e \in \partial(v_0)} p_e$ and $c_{v_0} := \max_{\boldsymbol{e} \in \mathcal{C}_{v_0}} |\boldsymbol{e}|$. Observe that if every probe involving an edge of $\partial(v_0)$ is inactive, then $\mathcal{M}(v_0) = \emptyset$. On the other hand, each probe independently fails with probability at least $(1 - p_{v_0})$, and there are at most $c_{v_0}$ probes made to $\partial(v_0)$. Thus,

$$\mathbb{P}[\mathcal{M}(v_0) = \emptyset \mid (\text{st}(e))_{e \in \widetilde{E}}, (Y_v)_{v \in V}] \geq (1 - p_{v_0})^{c_{v_0}}$$

Now, since $F(G) = (1 - 1/e) \cdot \min_{v \in V}(1 - p_v)^{c_v}$, we get that

$$\mathbb{E}[\alpha_{u_0} \mid (Y_v)_{v \in V}, (\text{st}(e))_{e \in \widetilde{E}}] \geq \left(1 - \frac{1}{e}\right)^{-1} w_{u_0}(1 - g(\widetilde{Y}_c)),$$

and so the proof is complete. □

By taking the appropriate conditional expectation, we can also lower bound the random variables $(\phi_{v_0,R})_{\substack{R \subseteq U: \\ u_0 \in R}}$.

**Proposition 7.15.** $\sum_{\substack{R \subseteq U: \\ u_0 \in R}} \mathbb{E}[\phi_{v_0,R} \mid (Y_v)_{v \in \widetilde{V}}, (st(e))_{e \in \widetilde{E}}] \geq \frac{1}{F} \int_0^{\widetilde{Y}_c} g(z)\, dz.$

*Proof of Proposition 7.15.* We first define $R_{v_0}$ as the unmatched vertices of $U$ when $v_0$ arrives (this is a random subset of $U$). We also once again use $\mathcal{M}$ to denote the matching returned by Algorithm 7 when executing on $G$. If we now take a *fixed* subset $R \subseteq U$, then the charging assignment to $\phi_{v_0,R}$ ensures that

$$\phi_{v_0,R} = w(\mathcal{M}(v_0)) \cdot \frac{g(Y_{v_0})}{F \cdot \text{OPT}(v_0, R)} \cdot \mathbf{1}_{[R_{v_0}=R]},$$

where $w(\mathcal{M}(v_0))$ corresponds to the weight of the vertex matched to $v_0$ (which is zero if $v_0$ remains unmatched after the execution on $G$). In order to make use of this relation, let us first condition on the values of $(Y_v)_{v \in V}$, as well as the states of the edges of $\widetilde{E}$; that is, $(\text{st}(e))_{e \in \widetilde{E}}$. Observe that once we condition on this information, we can determine $g(Y_{v_0})$, as well as $R_{v_0}$. As such,

$$\mathbb{E}[\phi_{v_0,R} \mid (Y_v)_{v \in V}, (\text{st}(e))_{e \in \widetilde{E}}] = \frac{g(Y_{v_0})}{F \cdot \text{OPT}(v_0, R)} \mathbb{E}[w(\mathcal{M}(v_0)) \mid (Y_v)_{v \in V}, (\text{st}(e))_{e \in \widetilde{E}}] \cdot \mathbf{1}_{[R_{v_0}=R]}.$$

On the other hand, the only randomness which remains in the conditional expectation involving $w(\mathcal{M}(v_0))$ is over the states of the edges adjacent to $v_0$. Observe now that since DP-OPT behaves optimally on $G[\{v_0\} \cup R_{v_0}]$, we get that

$$\mathbb{E}[w(\mathcal{M}(v_0)) \mid (Y_v)_{v \in V}, (\text{st}(e))_{e \in \widetilde{E}}] = \text{OPT}(v_0, R_{v_0}), \tag{7.14}$$

and so for the *fixed* subset $R \subseteq U$, $\mathbb{E}[w(\mathcal{M}(v_0)) \mid (Y_v)_{v \in V}, (\text{st}(e))_{e \in \widetilde{E}}] \cdot \mathbf{1}_{[R_{v_0}=R]} = \text{OPT}(v_0, R) \cdot \mathbf{1}_{[R_{v_0}=R]}$, after multiplying each side of (7.14) by the indicator random variable $\mathbf{1}_{[R_{v_0}=R]}$. Thus,

$$\mathbb{E}[\phi_{v_0,R} \mid (Y_v)_{v \in V}, (\text{st}(e))_{e \in \widetilde{E}}] = \frac{g(Y_{v_0})}{F} \mathbf{1}_{[R_{v_0}=R]},$$

after cancellation. We therefore get that $\sum_{\substack{R \subseteq U:\\ u_0 \in R}} \mathbb{E}[\phi_{v_0,R} \mid (Y_v)_{v \in V}, (\mathrm{st}(e))_{e \in \widetilde{E}}] = \frac{g(Y_{v_0})}{F} \sum_{\substack{R \subseteq U:\\ u_0 \in R}} \mathbf{1}_{[R_{v_0}=R]}$. Let us now focus on the case when $v_0$ arrives before the critical time; that is, $0 \leq Y_{v_0} < \widetilde{Y}_c$. Up until the arrival of $v_0$, the executions of the algorithm on $\widetilde{G}$ and $G$ proceed identically, thanks to the coupling between the executions. As such, $u_0$ must be available when $v_0$ arrives. We interpret this observation in the above notation as saying the following: $\mathbf{1}_{[Y_{v_0}<\widetilde{Y}_c]} \leq \sum_{\substack{R \subseteq U:\\ u_0 \in R}} \mathbf{1}_{[R_{v_0}=R]}$. As a result, $\sum_{\substack{R \subseteq U:\\ u_0 \in R}} \mathbb{E}[\phi_{v_0,R} \mid (Y_v)_{v \in V}, (\mathrm{st}(e))_{e \in \widetilde{E}}] \geq \frac{g(Y_{v_0})}{F} \mathbf{1}_{[Y_{v_0}<\widetilde{Y}_c]}$. Now, if we take expectation over $Y_{v_0}$, while still conditioning on the random variables $(Y_v)_{v \in \widetilde{V}}$, then we get that

$$\mathbb{E}[g(Y_{v_0}) \cdot \mathbf{1}_{[Y_{v_0}<\widetilde{Y}_c]} \mid (Y_v)_{v \in \widetilde{V}}, (\mathrm{st}(e))_{e \in \widetilde{E}}] = \int_0^{\widetilde{Y}_c} g(z)\, dz,$$

as $Y_{v_0}$ is drawn uniformly from $[0,1]$, independently from $(Y_v)_{v \in \widetilde{V}}$ and $(\mathrm{st}(e))_{e \in \widetilde{E}}$. Thus, after applying the law of iterated expectations,

$$\sum_{\substack{R \subseteq U:\\ u_0 \in R}} \mathbb{E}[\phi_{v_0,R} \mid (Y_v)_{v \in \widetilde{V}}, (\mathrm{st}(e))_{e \in \widetilde{E}}] \geq \frac{1}{F} \int_0^{\widetilde{Y}_c} g(z)\, dz,$$

and so the claim holds. □

With Propositions 7.13 and 7.15, the proof of Lemma 7.12 follows easily.

*Proof of Lemma 7.12.* We first observe that by taking the appropriate conditional expectation, Proposition 7.13 ensures that $\mathbb{E}[\alpha_{u_0} \mid (Y_v)_{v \in \widetilde{V}}, (\mathrm{st}(e))_{e \in \widetilde{E}}] \geq \left(1 - \frac{1}{e}\right)^{-1} w_{u_0} \cdot (1 - g(\widetilde{Y}_c))$, where the right-hand side follows since $\widetilde{Y}_c$ is entirely determined from $(Y_v)_{v \in \widetilde{V}}$ and $(\mathrm{st}(e))_{e \in \widetilde{E}}$. Thus, combined with Proposition 7.15,

$$\mathbb{E}[p_{u_0,v_0} \cdot \alpha_{u_0} + w_{u_0} \cdot p_{u_0,v_0} \cdot \sum_{\substack{R \subseteq U:\\ u_0 \in R}} \phi_{v,R} \mid (Y_v)_{v \in \widetilde{V}}, (\mathrm{st}(e))_{e \in \widetilde{E}}], \tag{7.15}$$

is lower bounded by

$$\left(1 - \frac{1}{e}\right)^{-1} w_{u_0} \cdot p_{u_0,v_0} \cdot (1 - g(\widetilde{Y}_c)) + \frac{w_{u_0}\, p_{u_0,v_0}}{F} \int_0^{\widetilde{Y}_c} g(z)\, dz. \tag{7.16}$$

However, $g(z) := \exp(z-1)$ for $z \in [0,1]$ by assumption, so $(1 - g(\widetilde{Y}_c)) + \int_0^{\widetilde{Y}_c} g(z)\, dz = \left(1 - \frac{1}{e}\right)$, no matter the value of the critical time $\widetilde{Y}_c$. Thus,

$$\left(1 - \frac{1}{e}\right)^{-1} \left((1 - g(\widetilde{Y}_c)) + \frac{1 - 1/e}{F} \int_0^{\widetilde{Y}_c} g(z)\, dz\right) \geq 1, \tag{7.17}$$

as $F \leq 1 - 1/e$ by definition (see (7.10)). If we now lower bound (7.16) using (7.17) and take expectations over (7.15), it follows that $\mathbb{E}[p_{u_0,v_0} \cdot \alpha_{u_0} + w_{u_0} \cdot p_{u_0,v_0} \cdot \sum_{\substack{R \subseteq U:\\ u_0 \in R}} \phi_{v,R}] \geq w_{u_0} \cdot p_{u_0,v_0}$. As the vertices $u_0 \in U$ and $v_0 \in V$ were chosen arbitrarily, the proposed dual solution of Lemma 7.12 is feasible, and so the proof is complete. □

## 8 Efficiency of Our Algorithms

In this section, we prove that all of our algorithms can be implemented efficiently in the membership oracle model. Given a stochastic graph $G$, we denote $|G|$ to be the number of bits needed to encode all of its parameters *excluding* its downward-closed probing constraints $(\mathcal{C}_v)_{v \in V}$. Clearly, Algorithm 7 has a runtime which is polynomial in $|G|$ by Theorem 7.1 (denoted poly($|G|$)). For our remaining algorithms, we first show that LP-config can be solved efficiently.

**Theorem 8.1.** *Suppose that $G = (U, V, E)$ in a stochastic graph with downward-closed probing constraints $(\mathcal{C}_v)_{v \in V}$. In the membership oracle model, LP-config can be solved in time poly($|G|$).*

We prove Theorem 8.1 by first considering the dual of LP-config. Note, that in the below LP formulation, if $\boldsymbol{e} = (e_1, \ldots, e_k) \in \mathcal{C}_v$, then we set $e_i = (u_i, v)$ for $i = 1, \ldots, k$ for convenience.

$$\begin{array}{llll}
\text{minimize} & & \displaystyle\sum_{u \in U} \alpha_u + \sum_{v \in V} \beta_v & \text{(LP-config-dual)} \\
\text{subject to} & \displaystyle\beta_v + \sum_{j=1}^{|\boldsymbol{e}|} p_{e_j} \cdot q(\boldsymbol{e}_{<j}) \cdot \alpha_{u_j} \geq & \displaystyle\sum_{j=1}^{|\boldsymbol{e}|} p_{e_j} \cdot w_{e_j} \cdot q(\boldsymbol{e}_{<j}) & \forall v \in V, \boldsymbol{e} \in \mathcal{C}_v \\
& & \alpha_u \geq 0 & \forall u \in U \\
& & \beta_v \in \mathbb{R} & \forall v \in V
\end{array}$$

Observe that to prove Theorem 8.1, it suffices to show that LP-config-dual has a (deterministic) polynomial time separation oracle, as a consequence of how the ellipsoid algorithm [45, 28] executes (see [48, 47, 2, 37] for more detail).

Suppose that we are presented a particular selection of dual variables, say $(\alpha_u)_{u \in U}$ and $(\beta_v)_{v \in V}$, which may or may not be a feasible solution to LP-config-dual. Our separation oracle must determine efficiently whether these variables satisfy all the constraints of LP-config-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 let us fix a particular $v \in V$ in what follows. We wish to determine whether there exists some $\boldsymbol{e} = (e_1, \ldots, e_k) \in \mathcal{C}_v$, such that if $e_i = (u_i, v)$ for $i = 1, \ldots, k$, then

$$\sum_{j=1}^{|\boldsymbol{e}|} (w_{e_j} - \alpha_{u_j}) \cdot p_{e_j} \cdot q(\boldsymbol{e}_{<j}) > \beta_v, \tag{8.1}$$

where the left-hand side of (8.1) is 0 if $\boldsymbol{e} = \lambda$. In order to make this determination, it suffices to solve the following maximization problem. Given any selection of real values, $(\alpha_u)_{u \in U}$,

$$\begin{array}{ll}
\text{maximize} & \displaystyle\sum_{i=1}^{|\boldsymbol{e}|} (w_{e_i} - \alpha_{u_i}) \cdot p_{e_i} \cdot \prod_{j=1}^{i-1} (1 - p_{e_j}) \\
\text{subject to} & \boldsymbol{e} \in \mathcal{C}_v
\end{array} \tag{8.2}$$

Before we show how (8.2) can be solved, we provide a buyer/seller interpretation of the optimization problem. Assuming first that the edges exist with certainty (i.e. $p_e \in \{0, 1\}$ for all $e \in \partial(v)$), let us suppose a seller is trying to allocate the items of $U$ to a number of buyers. We view the vertex $v$ as a *buyer* who wishes to purchase a subset of items $S \subseteq U$, based on their valuation function $f(S)$. Assume that $v$ has **unit demand**, that is $f(S) := \max_{s \in S} p_{s,v} w_{s,v}$. The values $(\alpha_s)_{s \in U}$ are viewed as prices the buyer must pay, and $\max_{S \subseteq U}(f(S) - \sum_{s \in S} \alpha_s)$ is the maximum utility of $v$. Clearly, for the simple case of a unit-demand buyer, an optimum assignment is the item $u \in U$ for which $p_{u,v} w_{u,v} - \alpha_u$ is maximized.

Returning the setting of arbitrary edge probabilities, even the case of a unit-demand buyer is a non-trivial optimization problem in the stochastic probing framework. Observe that we may view the edge probabilities $(p_e)_{e \in \partial(v)}$ as modelling the setting when there is uncertainty in whether or not the purchase proposals will succeed; that is, $\text{st}(u, v) = 1$, provided the seller agrees to sell item $u$ to buyer $v$. In this interpretation, (8.2) is the expected utility of the unit-demand buyer $v$ which purchases the first item $u \in U$ such that $\text{st}(u, v) = 1$, at which point $v$ gains utility $w_{u,v} - \alpha_u$. This easily reduces to the problem solved by DP-OPT, and we include the details below:

**Proposition 8.2.** *If $\mathcal{C}_v$ is downward-closed, then for any selection of values $(\alpha_u)_{u \in U}$,* (8.2) *can be solved efficiently by DP-OPT, assuming access to a membership query oracle for $\mathcal{C}_v$.*

*Proof.* Compute $\widetilde{w}_e := w_e - \alpha_u$ for each $e = (u, v) \in \partial(v)$, and define $P := \{e \in \partial(v) : \widetilde{w}_e \geq 0\}$. First observe that if $P = \emptyset$, then (8.1) is maximized by the empty-string $\lambda$. Thus, for now on assume that $P \neq \emptyset$. Since $\mathcal{C}_v$ is downward-closed, it suffices to consider those $\boldsymbol{e} \in \mathcal{C}_v$ whose edges all lie in $P$. As such, for notational convenience, let us hereby assume that $\partial(v) = P$. Observe then that solving (8.2) corresponds to executing DP-OPT on the stochastic graph $G[U \cup \{v\}]$, with edge weights replaced by $(\widetilde{w}_e)_{e \in \partial(v)}$. ☐

As a corollary of Theorem 8.1, it immediately follows that Algorithm 6 is poly-time in $|G|$. Let us now consider when we have a known i.d. instance $(H_{\text{typ}}, (\mathcal{D}_i)_{i=1}^n)$, and $|\mathcal{D}_i|$ denotes the amount of space needed to encode the distribution $\mathcal{D}_i$. Following the standard in the literature, we consider an algorithm to be efficient if it can be implemented in time $\text{poly}(|H_{\text{typ}}|, (|\mathcal{D}_i|)_{i=1}^n)$.

We first observe that LP-config-id can be solved in time $\text{poly}(|H_{\text{typ}}|, (|\mathcal{D}_i|)_{i=1}^n)$ using the same strategy as in Theorem 8.1, as the same maximization problem (8.2) is needed to separate the dual of LP-config-id. On the other hand, Theorem 5.4 is a polynomial-time reduction. Thus, since the OCRS and RCRS used in Corollary 5.5 and Corollary 5.6 are polynomial time, our $1/2$ and $1 - 1/e$ competitive ratios are attained by efficient online probing algorithms.

## 9 A Tight Adaptivity Gap

Let $G = (U, V, E)$ be a stochastic graph with **substring-closed** probing constraints $(\mathcal{C}_v)_{v \in V}$. Here $\mathcal{C}_v$ is substring-closed if any substring of $\boldsymbol{e} \in \mathcal{C}_v$ is also in $\mathcal{C}_v$. This is a less restrictive definition than imposing $\mathcal{C}_v$ must be downward-closed, and is the minimal assumption one needs to ensure that the offline stochastic matching is well-defined.

Given an offline probing algorithm, we say it is **non-adaptive** provided its probes are a randomized function of $G$. Similar to the definition of the offline adaptive benchmark, we define the **non-adaptive benchmark** as the optimum performance of a non-adaptive offline probing algorithm on $G$. That is, $\text{OPT}_{\text{n-adap}}(G) := \sup_{\mathcal{B}} \mathbb{E}[w(\mathcal{B}(G))]$, where the supremum is over all non-adaptive offline probing algorithms. We define the **adaptivity gap** of the **stochastic matching problem with one-sided probing constraints**, as the ratio

$$\inf_{G} \frac{\text{OPT}_{\text{n-adap}}(G)}{\text{OPT}(G)}, \tag{9.1}$$

where the infimum is over all (bipartite) stochastic graphs $G = (U, V, E)$ with substring-closed probing constraints $(\mathcal{C}_v)_{v \in V}$. Observe that (9.1) is upper bounded by 1 by definition. We now state a better upper bound (i.e., negative/impossibility result) on (9.1).

**Theorem 9.1.** *The adaptivity gap of the stochastic matching problem with one-sided probing constraints is upper bounded by $1 - 1/e$.*

Theorem 9.1 follows by considering a sequence of stochastic graphs. In particular, given $n \geq 1$, consider functions $p = p(n)$ and $s = s(n)$ which satisfy the following:

1. $p \ll 1/\sqrt{n}$ and $s \to \infty$ as $n \to \infty$.
2. $s \leq pn$ and $s = (1 - o(1))pn$.

Consider now an unweighted stochastic graph $G_n = (U, V, E)$ with unit patience values, and which satisfies $|U| = s$ and $|V| = n$. Moreover, assume that $p_{u,v} = p$ for all $u \in U$ and $v \in V$. Observe that $G_n$ corresponds to the bipartite Erdős–Rényi random graph $\mathbb{G}(s, n, p)$.

**Lemma 9.2.** *The offline adaptive benchmark returns a matching of size asymptotically equal to $s$ when executing on $G_n$; that is, $OPT(G_n) = (1 + o(1))s$.*

We omit the proof of Lemma 9.2, as it is routine analysis of the Erdős–Rényi random graph $\mathbb{G}(s, n, p)$. Instead, we focus on proving the following lemma, which together with Lemma 9.2 implies the upper bound of Theorem 9.1:

**Lemma 9.3.** *The non-adaptive benchmark returns in expectation a matching of size at most $(1 + o(1)) \left(1 - \frac{1}{e}\right) s$ when executing on $G_n$. That is,*

$$OPT_{n\text{-}adp}(G) \leq (1 + o(1)) \left(1 - \frac{1}{e}\right) s.$$

*Proof.* Let $\mathcal{A}$ be a non-adaptive probing algorithm, which we may assume is deterministic w.l.o.g. As the probes of $\mathcal{A}$ are determined independently of the random variables $(\mathrm{st}(e))_{e \in E}$, we can define $x_e \in \{0, 1\}$ for each $e \in E$ to indicate whether or not $\mathcal{A}$ probes the edge $e$.

Now, if $\mathcal{A}(G)$ is the matching returned by $\mathcal{A}$, then using the independence of the edge states $(\mathrm{st}(e))_{e \in E}$, we get that

$$\mathbb{P}[u \text{ matched by } \mathcal{A}(G)] \leq \mathbb{P}\left[\cup_{\substack{v \in V: \\ x_{u,v}=1}} \mathrm{st}(u, v) = 1\right] \tag{9.2}$$

$$= 1 - \prod_{v \in V} (1 - p x_{u,v}) \tag{9.3}$$

and so $\mathbb{E}[|\mathcal{A}(G))|] \leq s - \sum_{u \in U} \prod_{v \in V} (1 - p x_{u,v})$. As such, if we can show that

$$\sum_{u \in U} \prod_{v \in V} (1 - p x_{u,v}) \geq (1 - o(1)) \frac{s}{e},$$

then this will imply that $\mathbb{E}[|\mathcal{A}(G)|] \leq (1 + o(1)) \left(1 - \frac{1}{e}\right) s$.

To see this, first observe that since $p(n) \to 0$ as $n \to \infty$, we know that $1 - p x_{u,v} = (1 + o(1)) \exp(-p x_{u,v})$ for each $v \in V$. In fact, since $p x_{u,v} \leq p$ for all $v \in V$, the asymptotics are uniform across $V$. More precisely, there exists $C > 0$, such that for $n$ sufficiently large, $1 - p x_{u,v} \geq (1 - Cp^2) \exp(-p x_{u,v})$ for all $v \in V$. As a result,

$$\prod_{v \in V} (1 - p x_{u,v}) \geq (1 - Cp^2)^n \exp\left(-\sum_{v \in V} p x_{u,v}\right)$$

$$= (1 + o(1)) \exp\left(-\sum_{v \in V} p x_{u,v}\right),$$

where the second line follows since $p \ll 1/\sqrt{n}$ by assumption. On the other hand, Jensen's inequality ensures that

$$\sum_{u \in U} \frac{\exp\left(-\sum_{v \in V} p x_{u,v}\right)}{s} \geq \exp\left(-\frac{\sum_{u \in U, v \in V} p x_{u,v}}{n}\right).$$

However, $\sum_{u \in U} x_{u,v} \leq 1$ for each $v \in V$. Thus, $\sum_{u \in U, v \in V} p x_{u,v} \leq pn$, and so

$$\exp\left(-\frac{\sum_{u \in U, v \in V} p x_{u,v}}{s}\right) \geq \exp\left(-\frac{pn}{s}\right) \geq \frac{1}{e},$$

where the last line follows since $pn \leq s$. It follows that $\sum_{u \in U} \prod_{v \in V}(1 - p x_{u,v}) \geq (1 + o(1))\frac{s}{e}$, and so $\mathbb{E}[|\mathcal{A}(G)|] \leq (1 + o(1))\left(1 - \frac{1}{e}\right) s$. As the asymptotics hold uniformly across each deterministic non-adaptive algorithm $\mathcal{A}$, this completes the proof. $\square$

When considered in the known stochastic graph setting, the $1 - 1/e$-competitive algorithm of Corollary 5.6 is non-adaptive. Moreover, it applies more generally to stochastic graphs with substring-closed probing constraints. The stronger downward closed condition is only needed to solve LP-config efficiently. Thus, Corollary 5.6 and Theorem 9.1 exactly characterize the adaptivity gap of the offline stochastic matching problem with one-sided probing constraints:

**Corollary 9.4.** *The adaptivity gap of the offline stochastic matching problem with one-sided probing constraints is* $1 - 1/e$.

## 10 Conclusion and 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. [18] obtain a 0.669 competitive ratio following Azar et al. [6] who were the first to surpass the $1 - 1/e$ "barrier". Correa et al. [18] 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? As previously mentioned, Yan [49] recently proved that $0.645 > 1 - 1/e$ is attainable for known i.i.d. arrivals when probing is not required. 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 [46].

### Acknowledgements

We would like to thank Denis Pankratov, Rajan Udwani, and David Wajc for their constructive comments on early versions of this paper.

## 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] Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular stochastic probing on matroids. *Mathematics of Operations Research*, 41(3):1022–1038, 2016.

[5] 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, page 18–35, New York, NY, USA, 2012. Association for Computing Machinery.

[6] 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.

[7] 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.

[8] 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.

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

[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)*, volume 207 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:23, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[11] 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 Apr 2021.

[12] 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.

[13] Brian Brubach, Nathaniel Grammel, Will Ma, and Aravind Srinivasan. Online matching frameworks under stochastic rewards, product ranking, and unknown patience. *Operations Research*, 2023.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] Mahsa Derakhshan and Alireza Farhadi. Beating (1 - 1/e)-approximation for weighted stochastic matching. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 1931–1961. SIAM, 2023.

[21] 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.

[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, page 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, page 769–787, New York, NY, USA, 2020. Association for Computing Machinery.

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

[25] 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.

[26] 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.

[27] 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.

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

[29] Vineet Goyal and Rajan Udwani. Online matching with stochastic rewards: Optimal competitive ratio via path based formulation. In Péter Biró, Jason D. 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.

[30] Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In Michel X. Goemans and José R. Correa, editors, *Integer Programming and Combinatorial Optimization - 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings*, volume 7801 of *Lecture Notes in Computer Science*, pages 205–216. Springer, 2013.

[31] 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.

[32] Zhiyi Huang, Xinkai Shu, and Shuyi Yan. The power of multiple choices in online stochastic matching, 2022.

[33] 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.

[34] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for online 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.

[35] 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.

[36] Ulrich Krengel and Louis Sucheston. Semiamarts and finite values. *Bulletin of The American Mathematical Society - BULL AMER MATH SOC*, 83, 10 1977.

[37] 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.

[38] D.V. Lindley. Dynamic programming and decision theory. *Applied Statistics*, 10(1):39–51, 1961.

[39] Calum MacRury, Will Ma, and Nathaniel Grammel. On (random-order) online contention resolution schemes for the matching polytope of (bipartite) graphs. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 1995–2014. SIAM, 2023.

[40] 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.

[41] 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.

[42] Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2015:1388–1404, 10 2015.

[43] 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.

[44] Manish Purohit, Sreenivas Gollapudi, and Manish Raghavan. Hiring under uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5181–5189. PMLR, 09–15 Jun 2019.

[45] 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.

[46] 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, page 793–794, New York, NY, USA, 2021. Association for Computing Machinery.

[47] 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, page 783–792, New York, NY, USA, 2011. Association for Computing Machinery.

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

[49] Shuyi Yan. Edge-weighted online stochastic matching: Beating. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4631–4640. SIAM, 2024.

## A 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. [7], as well as LP-QC, the LP introduced by Gamlath et al. [26]. We then show that LP-QC and LP-config have the same optimum value when $G$ has unbounded patience.

**Theorem A.1.** *$OPT(G) = LPOPT(G)$*

*Proof.* Clearly, Theorem 3.1 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 3.2, 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 (3.1) of LP-config, and so the algorithm satisfies the required properties of a relaxed probing algorithm. The proof is therefore complete. $\square$

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 $e$.

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

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

$$\sum_{e \in \partial(v)} p_e \cdot x_e \leq 1 \qquad \forall v \in V \tag{A.2}$$

$$\sum_{e \in \partial(v)} x_e \leq \ell_v \qquad \forall v \in V \tag{A.3}$$

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

Observe that LP-config and LP-std are the same LP in the case of unit patience:

$$
\begin{aligned}
&\text{maximize} && \sum_{v \in V} \sum_{e \in \partial(v)} w_e \cdot p_e \cdot x_e && && \text{(LP-std-unit)} \\
&\text{subject to} && \sum_{e \in \partial(u)} p_e \cdot x_e \leq 1 && \forall u \in U && \text{(A.5)} \\
& && \sum_{e \in \partial(v)} x_e \leq 1 && \forall v \in V && \text{(A.6)} \\
& && x_e \geq 0 && \forall e \in E && \text{(A.7)}
\end{aligned}
$$

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

$$\sum_{e \in S} p_e \cdot x_e \leq 1 - \prod_{e \in S} (1 - p_e), \tag{A.8}$$

In the same variable interpretation as LP-std, the left-hand side of (A.8) 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[9].

$$
\begin{aligned}
&\text{maximize} && \sum_{e \in E} w_e \cdot p_e \cdot x_e && && \text{(LP-QC)} \\
&\text{subject to} && \sum_{e \in S} p_e \cdot x_e \leq 1 - \prod_{e \in S} (1 - p_e) && \forall v \in V, S \subseteq \partial(v) && \\
& && \sum_{e \in \partial(u)} p_e \cdot x_e \leq 1 && \forall u \in U && \text{(A.9)} \\
& && x_e \geq 0 && \forall e \in E. &&
\end{aligned}
$$

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

**Proposition A.2.** *If $G$ is unconstrained, then $LPOPT_{QC}(G) = LPOPT(G)$.*

In order to prove Proposition A.2, we make use of a result of Gamlath et al. We mention that an almost identical result is also proven by Costello et al. [19] using different techniques.

**Theorem A.3** ([26])**.** *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 $\mathbb{P}[\mathcal{B}_v \text{ matches } v \text{ to } e] = p_e x_e$ for each $e \in \partial(v)$.*

*Proof of Proposition A.2.* Observe that by Theorem A.1, 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. $\mathcal{M} \leftarrow \emptyset$.

[9]The LP considered by Gamlath et al. in [26] also places the analogous constraints of (A.8) 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.

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 A.3. If $\mathcal{B}_v$ matches $v$, then let $e'$ be this edge, and add $e'$ to $\mathcal{M}$

3. Return $\mathcal{M}$.

Using Theorem A.3, 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 (A.9). 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. □

## B Deferred Proofs

*Proof of Lemma 6.4.* Set $\alpha := 1/e$ for clarity, and take $t \geq \lceil \alpha n \rceil$. Define $e_t := (u_t, v_t)$, where $u_t$ is the vertex of $U$ which $v_t$ proposes to (which is the empty set $\emptyset$, if no such proposal occurs). For each $u \in U$, define $C(u, v_t)$ to be the event in which $v_t$ proposes to $u$. Let us now condition on the random subset $V_t$, as well as the random vertex $v_t$. In this case,

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

Observe however that once we condition on $V_t$ and $v_t$, Algorithm 6 corresponds to executing `VertexProbe` on the instance $(v_t, \partial(v_t), (x_{v_t}(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_v})$, where we recall that $(x_v(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_v, v \in v_t}$ is an optimum solution to LP-config for $G_t = G[U \cup V_t]$. Thus, $\mathbb{P}[C(u, v_t) \mid V_t, v_t] = p_{u,v_t} \widetilde{x}_{u,v_t}$ by Lemma 3.2, where

$$\widetilde{x}_{u,v_t} := \sum_{\substack{\boldsymbol{e}' \in \mathcal{C}_{v_t}: \\ e \in \boldsymbol{e}'}} q(\boldsymbol{e}'_{<e}) \cdot x_{v_t}(\boldsymbol{e}'),$$

and so $\mathbb{E}[w(e_t) \mid V_t, v_t] = \sum_{u \in U} w_{u,v_t} p_{u,v_t} \widetilde{x}_{u,v_t}$. 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 $(x_v(\boldsymbol{e}))_{v \in V_t, \boldsymbol{e} \in \mathcal{C}_v}$. These observations together imply that

$$\mathbb{E}[w_{u,v_t} \, p_{u,v_t} \, \widetilde{x}_{u,v_t} \mid V_t] = \frac{\sum_{v \in V_t} w_{u,v} \, p_{u,v} \, \widetilde{x}_{u,v}}{t} \tag{B.1}$$

for each $u \in U$ and $\lceil \alpha n \rceil \leq t \leq n$. If we now take expectation over $v_t$, then using the law of iterated expectations,

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

where the final equation follows from (B.1). Now, $\text{LPOPT}(G_t) = \sum_{v \in V_t} \sum_{u \in U} w_{u,v_t} \, p_{u,v_t} \, \widetilde{x}_{u,v_t}$, as $(x_v(\boldsymbol{e}))_{v \in V_t, \boldsymbol{e} \in \mathcal{C}_v}$ is an optimum solution to LP-config for $G_t$. As a result, $\mathbb{E}[w(e_t) \mid V_t] =$

LPOPT$(G_t)/t$, and so $\mathbb{E}[w(e_t)] = \mathbb{E}[\text{LPOPT}(G_t)]/t$, after taking taking expectation over $V_t$. On the other hand, Lemma 6.1 implies that

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

Thus, $\mathbb{E}[w(e_t)] \geq \text{LPOPT}(G)/n$, provided $\lceil \alpha n \rceil \leq t \leq n$, thereby completing the proof. □

*Proof of Lemma 6.5.* Let us assume that $\lceil \alpha n \rceil \leq t \leq n$ is fixed, and $(x_v^{(t)}(\boldsymbol{e}))_{v \in V, \boldsymbol{e} \in \mathcal{C}_v}$ is the optimum solution of LP-config for $G_t$, as used by Algorithm 6. For each $u \in U$ and $v \in v$, define the **edge variable** $\widetilde{x}_{u,v}^{(t)}$, where

$$\widetilde{x}_{u,v}^{(t)} := \sum_{\substack{\boldsymbol{e}' \in \mathcal{C}_{v_t}: \\ e \in \boldsymbol{e}'}} q(\boldsymbol{e}'_{<e}) \cdot x_{v_t}^{(t)}(\boldsymbol{e}')$$

We wish to prove that for each $u \in U$,

$$\mathbb{P}[u \in R_t \,|\, V_t, v_t] \geq \lfloor \alpha n \rfloor / (t-1). \tag{B.2}$$

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 $j = 1, \ldots, n$, denote $C(u, v_j)$ as the event in which $v_j$ proposes to $u \in U$. As $R_t$ denotes the unmatched nodes after the vertices $v_1, \ldots, v_{t-1}$ are processed by Algorithm 6, observe that $u \in R_t$ if and only if $\neg C(u, v_j)$ occurs for each $j = 1, \ldots, t-1$, and so $\mathbb{P}[u \in R_t] = \mathbb{P}[\cap_{j=1}^{t-1} \neg C(u, v_j)]$. 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 $j = 1, \ldots, \lfloor \alpha n \rfloor$, the algorithm passes on probing $\partial(v_j)$ by definition, and so (B.2) holds if $t = \lceil \alpha n \rceil$. We may thereby assume that $t \geq \lceil \alpha n \rceil + 1$ and focus on lower bounding $\mathbb{P}[\cap_{j=\lceil \alpha n \rceil}^{t-1} \neg C(u, v_j)]$. Observe that this event depends only on the probes of the vertices of $V_{t-1} \setminus V_{\lfloor \alpha n \rfloor}$. We denote $\bar{t} := t - 1 - \lfloor \alpha n \rfloor = t - \lceil \alpha n \rceil$ as the number of vertices within this set.

Let us first consider the vertex $v_{t-1}$, and the edge variable $\widetilde{x}_{u,v}^{(t-1)}$ for each $v \in V_{t-1}$. Observe that after applying Lemma 3.2,

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

as once we condition on the set $V_t$ and the vertex $v_t$, we know that $v_{t-1}$ 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-config for $G_{t-1}$, and so by constraint (3.1), $\sum_{v \in V_{t-1}} \widetilde{x}_{u,v}^{(t-1)} p_{u,v} \leq 1$. 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 $V_{t-1} \setminus V_{\lfloor \alpha n \rfloor}$. 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 proposals they (potentially) made. Formally, $\mathcal{H}_k$ is the sigma-algebra generated from $V_t, v_t$ and $\bar{e}_1, \ldots, \bar{e}_k$. By convention, we define $\mathcal{H}_0$ as the sigma-algebra generated from $V_t$ and $v_t$.

An analogous computation to the above case then implies that

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

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

Observe now that in each step, we condition on strictly more information; that is, $\mathcal{H}_{k-1} \subsetneq \mathcal{H}_k$ 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:

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

It follows that if $k = \bar{t} = t - \lceil \alpha n \rceil$, then applying the above recursion implies that

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

Thus, after cancelling the pairwise products, $\mathbb{P}[u \in R_t] = \mathbb{P}[\cap_{j=\lceil \alpha n \rceil}^{t-1} \neg C(u, v_j)] \geq \frac{\lfloor \alpha n \rfloor}{t-1}$, and so (B.2) holds for all $t \geq \lceil \alpha n \rceil$, thereby completing the argument. □

*Proof of Lemma 7.4.* We must show that for each fixed $u_0 \in U$ and $v_0 \in V$,

$$\mathbb{E}[p_{u_0,v_0} \cdot \alpha_{u_0} + w_{u_0} \cdot p_{u_0,v_0} \sum_{\substack{R \subseteq U: \\ u_0 \in R}} \phi_{v_0,R}] \geq w_{u_0} \cdot p_{u_0,v_0}. \tag{B.3}$$

Let $R_{v_0}$ be the unmatched vertices of $U$ when $v_0$ arrives (this is a random subset of $U$). Moreover, define $\widetilde{E} := E \setminus \partial(v_0)$. We claim the following:

$$\sum_{\substack{R \subseteq U: \\ u_0 \in R}} \mathbb{E}[\phi_{v_0,R} \mid (\mathrm{st}(e))_{e \in \widetilde{E}}] = \mathbf{1}_{[u_0 \in R_{v_0}]}.$$

To see this, observe that if we take a *fixed* subset $R \subseteq U$, then the charging assignment to $\phi_{v_0,R}$ ensures that

$$\phi_{v_0,R} = w(\mathcal{M}(v_0)) \cdot \frac{1}{\mathrm{OPT}(v_0, R)} \cdot \mathbf{1}_{[R_{v_0} = R]},$$

where $w(\mathcal{M}(v_0))$ corresponds to the weight of the vertex matched to $v_0$ (which is zero if $v_0$ remains unmatched after the execution on $G$). In order to make use of this relation, let us first condition on $(\mathrm{st}(e))_{e \in \widetilde{E}}$. Observe that once we condition on this information, we can determine $R_{v_0}$. As such,

$$\mathbb{E}[\phi_{v_0,R} \mid (\mathrm{st}(e))_{e \in \widetilde{E}}] = \frac{1}{\mathrm{OPT}(v_0, R)} \mathbb{E}[w(\mathcal{M}(v_0)) \mid (\mathrm{st}(e))_{e \in \widetilde{E}}] \cdot \mathbf{1}_{[R_{v_0} = R]}.$$

On the other hand, the only randomness which remains in the conditional expectation involving $w(\mathcal{M}(v_0))$ is over $(\text{st}(e))_{e \in \partial(v_0)}$. However, since Algorithm 7 behaves optimally on $G[\{v_0\} \cup R_{v_0}]$, we get that

$$\mathbb{E}[w(\mathcal{M}(v_0)) \,|\, (\text{st}(e))_{e \in \widetilde{E}}] = \text{OPT}(v_0, R_{v_0}), \tag{B.4}$$

and so for the *fixed* subset $R \subseteq U$, $\mathbb{E}[w(\mathcal{M}(v_0)) \,|\, (\text{st}(e))_{e \in \widetilde{E}}] \cdot \mathbf{1}_{[R_{v_0} = R]} = \text{OPT}(v_0, R) \cdot \mathbf{1}_{[R_{v_0} = R]}$ after multiplying each side of (B.4) by the indicator random variable $\mathbf{1}_{[R_{v_0} = R]}$. Thus, $\mathbb{E}[\phi_{v_0, R} \,|\, (\text{st}(e))_{e \in \widetilde{E}}] = \mathbf{1}_{[R_{v_0} = R]}$, after cancellation. We therefore get that $\sum_{\substack{R \subseteq U: \\ u_0 \in R}} \mathbb{E}[\phi_{v_0, R} \,|\, (\text{st}(e))_{e \in \widetilde{E}}] = \sum_{\substack{R \subseteq U: \\ u_0 \in R}} \mathbf{1}_{[R_{v_0} = R]} = \mathbf{1}_{[u_0 \in R_{v_0}]}$, as claimed. On the other hand, if we focus on the vertex $u_0$, then observe that if $u_0 \notin R_{v_0}$, then $\alpha_{u_0}$ must have been charged $w_{u_0}$. In other words, $\alpha_{u_0} \geq w_{u_0} \cdot \mathbf{1}_{[u_0 \notin R_{v_0}]}$. As a result,

$$\mathbb{E}[p_{u_0, v_0} \alpha_{u_0} + w_{u_0} p_{u_0, v_0} \sum_{\substack{R \subseteq U: \\ u_0 \in R}} \phi_{v,R} \,|\, (\text{st}(e))_{e \in \widetilde{E}}] \geq w_{u_0} p_{u_0, v_0} \cdot \mathbf{1}_{[u_0 \notin R_{v_0}]} + w_{u_0} p_{u_0, v_0} \cdot \mathbf{1}_{[u_0 \in R_{v_0}]},$$

and so (B.3) follows after taking expectations. The solution $((\alpha_u^*)_{u \in U}, (\phi_{v,R}^*)_{v \in V, R \subseteq U})$ is therefore feasible. □