Secretary Matching Meets Probing with Commitment

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

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

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

Abstract

We consider the online bipartite matching problem within the context of stochastic probing with commitment. This is the one-sided online bipartite matching problem where edges adjacent to an online node must be probed to determine if they exist based on edge probabilities that become known when an online vertex arrives. If a probed edge exists, it must be used in the matching. We consider the competitiveness of online algorithms in the adversarial order model (AOM) and the secretary/random order model (ROM). More specifically, we consider an unknown bipartite stochastic graph $G = (U, V, E)$ where U is the known set of offline vertices, V is the set of online vertices, G has edge probabilities $(p_e)_{e \in E}$, and G has edge weights $(w_e)_{e \in E}$ or vertex weights $(w_u)_{u \in U}$. Additionally, G has a downward-closed set of probing constraints $(\mathcal{C}_v)_{v \in V}$, where $\mathcal{C}_v$ indicates which sequences of edges adjacent to an online vertex v can be probed. This model generalizes the various settings of the classical bipartite matching problem (i.e. with and without probing). Our contributions include the introduction and analysis of probing within the random order model, and our generalization of probing constraints which includes budget (i.e. knapsack) constraints. Our algorithms run in polynomial time assuming access to a membership oracle for each $\mathcal{C}_v$.

In the vertex weighted setting, for adversarial order arrivals, we generalize the known $\frac{1}{2}$ competitive ratio to our setting of $\mathcal{C}_v$ constraints. For random order arrivals, we show that the same algorithm attains an asymptotic competitive ratio of $1 - 1/e$, provided the edge probabilities vanish to 0 sufficiently fast. We also obtain a strict competitive ratio for non-vanishing edge probabilities when the probing constraints are sufficiently simple. For example, if each $\mathcal{C}_v$ corresponds to a patience constraint ℓ_v (i.e., ℓ_v is the maximum number of probes of edges adjacent to v), and any one of following three conditions is satisfied (each studied in previous papers), then there is a conceptually simple greedy algorithm whose competitive ratio is $1 - \frac{1}{e}$.

- When the offline vertices are unweighted.
- When the online vertex probabilities are "vertex uniform"; i.e., $p_{u,v} = p_v$ for all $(u, v) \in E$.
- When the patience constraint ℓ_v satisfies $\ell_v \in \{[1, |U|\}$ for every online vertex; i.e., every online vertex either has unit or full patience.

Finally, in the edge weighted case, we match the known optimal $\frac{1}{e}$ asymptotic competitive ratio for the classic (i.e. without probing) secretary matching problem.

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

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

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2021.13

Category APPROX

Related Version *Full Version*: https://arxiv.org/abs/2008.09260 [4]

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 possible stochastic uncertainty in the set of inputs, stochastic probing problems involve inputs that cannot be determined without probing (at some cost and/or within some constraint) so as to reveal the inputs. Applications of stochastic probing occur naturally in many settings, such as in matching problems where compatibility (for example, in online dating and kidney exchange applications) or legality (for example, a financial transaction that must be authorized before it can be completed) cannot be determined without some trial or investigation. Amongst other applications, the online bipartite stochastic matching problem notably models online advertising where the probability of an edge can correspond to the probability of a purchase in online stores or to pay-per-click revenue in online searching. Commitment reflects the fact that one usually chooses the next probe based on some concept of expected value but in many applications (e.g. kidney exchanges) the cost or invasiveness of probing makes it practically necessary to commit. In some applications, there may be a legal requirement to commit (e.g., if a contract is possibly being offered and commitment is required).

The *(offline) stochastic matching* problem was introduced by Chen et al. [9]. In this problem, the input is an adversarially generated *stochastic graph* $G = (V, E)$ with a probability p_e associated with each edge e and a patience (or time-out) parameter ℓ_v associated with each vertex v. An algorithm probes edges in E within the constraint that at most ℓ_v edges are probed incident to any particular vertex $v \in V$. Also, when an edge e is probed, it is guaranteed to exist with probability exactly p_e. If an edge (u, v) is found to exist, it is added to the matching and then u and v are no longer available. The goal is to maximize the expected size of a matching constructed in this way. Chen et al. showed that by probing edges in non-increasing order of edge probability, one attains an approximation ratio of 1/4. The analysis was later improved by Adamczyk [1], who showed that this algorithm in fact attains an approximation ratio of 1/2. This problem can be generalized to vertices or edges having weights.

Mehta and Panigrahi [22] adapted the offline stochastic matching model to online bipartite matching as originally studied in the classical (non-stochastic) adversarial order online model. That is, they consider the setting where the stochastic graph is unknown and online vertices are determined by an adversary. More specifically, they studied the problem in the case of an unweighted stochastic graph $G = (U, V, E)$ where U is the set of known offline vertices and the vertices in V arrive online without knowledge of future online node arrivals. They considered the special case of uniform edge probabilities (i.e, $p_e = p$ for all $e \in E$) and *unit* patience values, that is $\ell_v = 1$ for all $v \in V$. They considered a greedy algorithm which attains a competitive ratio of $\frac{1}{2}(1 + (1 - p)^{2/p})$, which limits to $\frac{1}{2}(1 + e^{-2}) \approx .567$ as $p \to 0$. Mehta et al. [23] considered the unweighted online stochastic bipartite setting with arbitrary edge probabilities, attaining a competitive ratio of 0.534, and recently, Huang and Zhang [16] additionally handled the case of arbitrary offline vertex weights, while improving this ratio to 0.572. However, as in [22], both [23] and [16] are restricted to unit patience values, and moreover require edge probabilities which are *vanishingly small*[1]. Goyal and Udwani [12] improved on both of these works by showing a 0.596 competitive ratio in the same setting.

[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.

In all our results we will assume *commitment*; that is, when an edge is probed and found to exist, it must be included in the matching (if possible without violating the matching constraint). The patience constraint can be viewed as a simple form of a budget (equivalently, knapsack) constraint for the online vertices. We generalize patience and budget constraints by associating a downward-closed set $\mathcal{C}_v$ of probing sequences for each online node v where $\mathcal{C}_v$ indicates which sequences of edges adjacent to vertex v can be probed. In the general query and commit framework of Gupta and Nagarajan [14], the $\mathcal{C}_v$ constraints are the outer constraints.

1.1 Preliminaries

An input to the **(online) stochastic matching problem** is a **(bipartite) 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 $(p_e)_{e \in E}$. We draw an independent Bernoulli random variable of parameter p_e for each $e \in E$. We refer to this Bernoulli as the **state** of the edge e, and denote it by $\text{st}(e)$. If $\text{st}(e) = 1$, then we say that e is **active**, and otherwise we say that e is **inactive**. For each $v \in V$, denote $\partial(v)$ as the edges of G which include v. 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 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 $\boldsymbol{e}$. Thus, in particular, our setting encodes the case when v has a patience value ℓ_v, and more generally, when $\mathcal{C}_v$ corresponds to a matroid or budgetary constraint[2] on $\partial(v)$. Note that we will often assume w.l.o.g. that $E = U \times V$, as we can always set $p_{u,v} := 0$.

A solution to the online stochastic matching problem is an **online probing algorithm**. An online probing algorithm is initially only aware of the identity of the offline vertices U of G. We think of G, as well as the relevant edges probabilities, weights, and probing constraints, as being generated by an adversary. An ordering on V is then generated either through an adversarial process or uniformly at random. We refer to the former case as the **adversarial order model (AOM)** and the latter case as the **random order model (ROM)**.

Based on whichever ordering is generated on V, the nodes are then presented to the online probing algorithm one by one. When an online node $v \in V$ arrives, the online probing algorithm sees all the adjacent edges and their associated probabilities, as well as $\mathcal{C}_v$. However, the edge states $(\text{st}(e))_{e \in \partial(v)}$ remain hidden to the algorithm. Instead, the algorithm must perform a **probing operation** on an adjacent edge e to reveal/expose its state, $\text{st}(e)$. Moreover, the online probing algorithm must **respect commitment**. That is, if an edge $e = (u, v)$ is probed and turns out to be active, then e must be added to the current matching, provided u and v are both currently unmatched. The probing constraint $\mathcal{C}_v$ of the online node then restricts which sequences of probes can be made to $\partial(v)$. As in the classical problem, an online probing algorithm must decide on a possible match for an online node v before seeing the next online node. The goal of the online probing algorithm is to return a matching whose expected weight is as large as possible. Since $\mathcal{C}_v$ may be exponentially large in the size of U, in order to discuss the efficiency of an online probing algorithm, we work in the **membership oracle model**. That is, upon receiving the online

[2] In the case of a budget B_v and edge probing costs $(b_e)_{e \in \partial(v)}$, any subset of $\partial(v)$ may be probed, provided its cumulative cost does not exceed B_v.

vertex $v \in V$, we assume the online probing algorithm has access to **a membership oracle**. The algorithm may **query** any string $\boldsymbol{e} \in \partial(v)^{(*)}$, thus determining in a single operation whether or not $\boldsymbol{e} \in \partial(v)^{(*)}$ is in $\mathcal{C}_v$.

It is easy to see we cannot hope to obtain a non-trivial competitive ratio against the expected value of an optimal matching of the stochastic graph. Consider 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 present. The expectation of an online probing algorithm will be at most $\frac{1}{k}$ while the expected size of an optimal matching will be $1-(1-\frac{1}{k})^k \to 1-\frac{1}{e}$. This example clearly shows that no constant ratio is possible if the patience is sublinear in $k = |U|$. Thus, the standard in the literature is to instead benchmark the performance of an online probing algorithm against an *optimal offline probing algorithm*. An **offline probing algorithm** knows $G = (U, V, E)$, but initially the edge states $(\mathrm{st}(e))_{e \in E}$ are hidden. It can adaptively probe the edges of E in any order, but must satisfy the probing constraints $(\mathcal{C}_v)_{v \in V}$ at each step of its execution[3], while respecting commitment; that is, if a probed edge $e = (u, v)$ turns out to be active, then e is added to the matching (if possible). The goal of an offline probing algorithm is to construct a matching with optimal weight in expectation. We define the **committal benchmark** $\mathrm{OPT}(G)$ for G as the value of an optimal offline probing algorithm. We abuse notation slightly, and also use $\mathrm{OPT}(G)$ to refer to the *strategy* of the committal benchmark on G. In the arXiv version of the paper [4], we introduce the stronger **non-committal benchmark**, and indicate which of our results hold against it.

1.2 Our Results

We first consider the case when the stochastic graph $G = (U, V, E)$ has **(offline) vertex weights** – i.e., there exists $(w_u)_{u \in U}$ such that $w_{u,v} = w_u$ for each $v \in N(u)$, and arbitrary downward-closed probing constraints $(\mathcal{C}_v)_{v \in V}$. We consider a *greedy* online probing algorithm. That is, upon the arrival of v, the probes to $\partial(v)$ are made in such a way that v gains as much value 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 the committal benchmark when restricted to the **induced stochastic graph**[4] $G[\{v\} \cup R]$, which we denote by $\mathrm{OPT}(R, v)$ for convenience.

Observe that if v has unit patience, then $\mathrm{OPT}(R, v)$ reduces to probing the adjacent edge $(u, v) \in R \times \{v\}$ such that the value $w_u \cdot p_{u,v}$ is maximized. Moreover, if v has unlimited patience, then $\mathrm{OPT}(R, v)$ corresponds to probing the adjacent edges of $R \times \{v\}$ in non-increasing order of the associated vertex weights. Building on a result in Purohit et al. [24], Brubach et al. [8] showed how to devise an *efficient* probing strategy for v whose expected value matches $\mathrm{OPT}(R, v)$, for any patience value. Using this probing strategy, they devised an online probing algorithm which achieves a competitive ratio of 1/2 for arbitrary patience values. The challenge in extending this competitive ratio to more general probing constraints comes from the fact that it is unclear how to compute $\mathrm{OPT}(R, v)$ efficiently. We show that this is possible to do when the probing constraints are downward-closed, and provide a primal-dual proof of the following theorem:

[3] Edges $\boldsymbol{e} \in E^{(*)}$ may be probed in the order specified by $\boldsymbol{e}$, provided $\boldsymbol{e}^v \in \mathcal{C}_v$ for each $v \in V$, where $\boldsymbol{e}^v$ is the substring of $\boldsymbol{e}$ restricted to edges of $\partial(v)$.

[4] 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 $\mathcal{C}_v$ is restricted to those strings whose entries lie entirely in $R \times \{v\}$.

▶ **Theorem 1.1.** *Suppose the adversary presents a vertex weighted stochastic graph $G = (U, V, E)$, with downward-closed probing constraints $(\mathcal{C}_v)_{v \in V}$. If $\mathcal{M}$ is the matching returned by Algorithm 1 when executing on G, then*

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

provided the vertices of V arrive in adversarial order. Moreover, Algorithm 1 can be implemented efficiently in the membership oracle model.

Since Algorithm 1 is deterministic, the 1/2 competitive ratio is best possible for deterministic algorithms in the adversarial arrival setting. One direction is thus to instead consider what can be done if the online probing algorithm is allowed randomization, which has received much attention in the case of unit patience [22, 23, 12, 16]. We instead make partial progress to understanding the performance of Algorithm 1 for downward-closed probing constraints in the ROM setting. However, unlike the adversarial setting, the complexity of the constraints greatly impacts what we are able to prove. The first part of our result is asymptotic in that it yields a good competitive ratio when applied to a stochastic graph whose maximum edge probability $p_v := \max_{e \in \partial(v)} p_e$ vanishes sufficiently fast relevant to the maximum string length of $\mathcal{C}_v$, namely $c_v := \max_{\boldsymbol{e} \in \mathcal{C}_v} |\boldsymbol{e}|$, for each $v \in V$. Note that the vanishing probability setting is similar in spirit to the small bid to budget assumption in the Adwords problem (see Goyal and Udwani [12] for details). The second part of our result applies to stochastic graphs which we refer to as **rankable**. Roughly speaking, a vertex $v \in V$ of G is **rankable**, provided there exists a fixed/non-adaptive ranking π_v of $\partial(v)$ which can be used to specify the greedy strategy $\text{OPT}(v, R)$ of v, no matter which vertices $R \subseteq U$ are available when v is processed. For example, this includes the well-studied unit patience setting, in which case v ranks its adjacent edges in non-increasing order of $(w_u p_{u,v})_{u \in U}$, as well as when G is unweighted and has arbitrary patience values, in which case v ranks its adjacent edges in non-increasing order of edge probability. A stochastic graph is rankable if all its online vertices are rankable. We defer the precise definition to Section 2.

▶ **Theorem 1.2.** *Suppose Algorithm 1 returns the matching $\mathcal{M}$ when executing on the vertex weighted stochastic graph $G = (U, V, E)$ with downward-closed constraints $(\mathcal{C}_v)_{v \in V}$, and the vertices of V arrive u.a.r.. We then have the following two results:*

1. *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 $c_v \cdot p_v \to 0$ (as $|G| \to \infty$) for each $v \in V$, then $\mathbb{E}[w(\mathcal{M})] \geq (1 - o(1))\,(1 - 1/e) \cdot OPT(G)$.

2. *If G is rankable (which includes the specific cases outlined in the abstract), then*

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

▶ Remark 1.3. The analysis of Algorithm 1 is tight, as an execution of Algorithm 1 corresponds to the seminal Karp et al. [17] Ranking algorithm for unweighted non-stochastic (i.e., $p_e \in \{0, 1\}$ for all $e \in E$) bipartite matching.

In the unit patience setting of [22], Mehta and Panigrahi showed that $.621 < 1 - \frac{1}{e}$ is a randomized inapproximation with regard to guarantees made against LP-std-unit, the LP introduced by [22] to upper bound/relax the committal benchmark in the unit patience setting. This hardness result led Goyal and Udwani [12] to consider a new unit patience LP that is a tighter relaxation of OPT(G) than LP-std-unit, thereby allowing them to prove a $1 - 1/e$ competitive ratio for the case of **vertex-decomposable**[5] edge probabilities. However, they also discuss the difficulty of extending this result to the case of arbitrary edge probabilities in the context of the Adwords problem with arbitrary budget to bid ratios. It remains open whether a randomized algorithm can attain a competitive ratio of $1 - 1/e$ against the committal benchmark for adversarial arrivals and arbitrary edge probabilities. A corollary of Theorem 1.2 is that in the ROM setting these difficulties do *not* arise.

▶ **Corollary 1.4.** *Suppose the adversary presents a vertex weighted stochastic graph $G = (U, V, E)$, with unit patience values. If $\mathcal{M}$ is the matching returned by Algorithm 1 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 1.5. The guarantee of Theorem 1.2 is proven against a new LP relaxation (LP-DP) whose optimum value we denote by $\text{LPOPT}_{\text{DP}}(G)$. In the special case when G has unit patience, $\text{LPOPT}_{\text{std}}(G) \leq \text{LPOPT}_{\text{DP}}(G)$. Thus, the 0.621 inapproximation of Mehta and Panigraphi against LP-std-unit does not apply (even for deterministic probing algorithms) to the ROM setting. Corollary 1.4 therefore implies that deterministic probing algorithms in the ROM setting have strictly more power than randomized probing algorithms in the adversarial order model. This contrasts with the classic ROM setting where it is unknown whether a deterministic algorithm can improve upon $1 - 1/e$, the optimal competitive attainable by randomized algorithms in the adversarial setting.

We next consider the unknown stochastic matching problem in the most general setting of arbitrary edge weights, and downward-closed probing constraints. Since no non-trivial competitive ratio can be proven in the case of adversarial arrivals, even in the classical setting, we work in the ROM setting. We generalize the matching algorithm of Kesselheim et al. [18] so as to apply to the stochastic probing setting.

▶ **Theorem 1.6.** *Suppose the adversary presents an edge-weighted stochastic graph $G = (U, V, E)$, with downward-closed probing constraints $(\mathcal{C}_v)_{v \in V}$. If $\mathcal{M}$ is the matching returned by Algorithm 2 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 uniformly at random (u.a.r.). Moreover, Algorithm 2 can be implemented efficiently in the membership oracle model.

▶ Remark 1.7. For context, the previous best known approximation ratio known for the offline bipartite stochastic matching problem with two-sided or one-sided patience is 0.352 due to Adamczyk et al. [3]. Since $1/e > 0.352$, Theorem 1.6 in fact improves on this result for the case of one-sided patience, despite the fact that Algorithm 2 works in the unknown graph setting and for more general one-sided probing constraints. Very recently, Brubach et al. [7] proved an approximation ratio of 0.382 for general stochastic graphs.

[5] 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 $(u, v) \in E$.

1.3 Our Technical Contributions

In the vertex weighted setting, the first challenge is to establish a greedy strategy for a single online vertex which runs efficiently for general probing constraints. We provide a dynamic programming based algorithm (DP-OPT) for solving this problem, which builds upon the work of Brubach et al. [8], and before that, Purohit et al. [24] (see Theorem 2.1). In the adversarial arrival setting, we prove a competitive ratio of 1/2 by comparing the performance of Algorithm 1 to the dual of LP-DP, an extension of the LP considered by Brubach et al. [8] from patience values to general probing constraints.

We next move to the ROM/secretary setting. In the unit patience setting of Corollary 1.4, DP-OPT reduces to probing a single edge which yields the largest value in expectation, and LP-DP is a relaxation of LP-std-unit (upper bounds its optimum value). While we do not show this, one could work directly with LP-std-unit and follow the primal-dual argument of Devanur et al. [10]. In contrast, Theorem 1.2 applies to downward-closed probing constraints which comes with two main technical challenges. First, Brubach et al. [8] showed that even the offline committal benchmark has a 0.544 inapproximation against the generalization of LP-std-unit to arbitrary patience (LP-std). Moreover, this inapproximation applies to a stochastic graph which is both rankable and has vanishingly small edge probabilities. Thus, Theorem 1.2 cannot be proven by comparing the performance of Algorithm 1 to LP-std and its dual, even for patience values. Our solution is to instead work with LP-DP and its dual, LP-dual-DP. When a match between $u \in U$ and $v \in V$ is successfully made, we apply the well-studied cost sharing function $g(z) := \exp(z-1)$ to split the weight of u, as in [10]. However, LP-dual-DP contains variables which do *not* have an analogue in the classical setting. Specifically, the online vertices are associated with exponentially many variables, and we cost share with the offline vertices which were available when v was matched to u, opposed to just v itself. The second main technical challenge is that when moving away from the unit patience setting, the executions of Algorithm 1 become **non-monotonic**. Specifically, while v may get matched to u, if a new online vertex v^* is added to the graph ahead of v, then u may not be matched at all. This complicates the analysis, and is the reason the competitive ratio of Theorem 1.2 does not hold unconditionally, as we explain in Section 2.

In the edge weighted setting, upon receiving the online vertices $V_t := \{v_1, \ldots, v_t\}$, in order to generalize the matching algorithm of Kesselheim et al. [18], Algorithm 2 would ideally probe the edges of $\partial(v_t)$ suggested by $\mathrm{OPT}(G_t)$, 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 make use of a new LP (LP-config) recently introduced by the authors in [5] and independently by Brubach et al. in [6, 13] for the special case of patience values, an updated version of [8]. This LP has exponentially many variables which accounts for the many probing strategies available to an arriving vertex v with probing constraint $\mathcal{C}_v$. We solve this LP efficiently by using DP-OPT as a deterministic separation oracle for LP-config-dual, the dual of LP-config, in conjunction with the ellipsoid algorithm [26, 11]. This LP closely resembles what the committal benchmark is capable of doing, and thus leads to a probing algorithm with an optimum competitive ratio.

2 Vertex Weights

In this section, we define Algorithm 1 and introduce the techniques needed to prove Theorems 1.1 and 1.2. However, for space considerations, we defer the dual-fitting argument used in the adversarial arrival setting of Theorem 1.1 to Appendix B.

Suppose that $G = (U, V, E)$ is a vertex weighted stochastic graph with weights $(w_u)_{u \in U}$. Let us now fix $s \in V$, and define $\text{val}(\boldsymbol{e})$ to be the expected weight of the edge matched, provided the edges of $\boldsymbol{e}$ are probed in order, where $\boldsymbol{e} \in \mathcal{C}_s$. Observe then the following claim:

▶ **Theorem 2.1.** *There exists a dynamic programming (DP) based algorithm* DP-OPT*, which given access to $G[\{s\} \cup U]$, computes a tuple $\boldsymbol{e}' \in \mathcal{C}_s$, such that $OPT(s, U) = val(\boldsymbol{e}')$. Moreover,* DP-OPT *executes in time $O(|U|^2)$, assuming access to a membership oracle for the downward-closed constraint $\mathcal{C}_s$.*

Proof of Theorem 2.1. It will be convenient to denote $w_{u,s} := w_u$ for each $u \in U$ such that $(u, s) \in \partial(s)$. We first must show that there exists some $\boldsymbol{e}' \in \mathcal{C}_s$ such that $\text{val}(\boldsymbol{e}') = \text{OPT}(s, U)$, where

$$\text{val}(\boldsymbol{e}) := \sum_{i=1}^{|\boldsymbol{e}|} p_{e_i} w_{e_i} \prod_{j=1}^{i-1} (1 - p_{e_i}), \tag{2.1}$$

for $\boldsymbol{e} \in \mathcal{C}_s$, and $\text{OPT}(s, U)$ is the value of the committal benchmark on $G[\{s\} \cup U]$. Since the committal benchmark must respect commitment – i.e., match the first edge to s which it reveals to be active – it is clear that $\boldsymbol{e}'$ exists.

Our goal is to now 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 [24, 8]). 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 (2.1) 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 $f : 2^{\partial(s)} \to [0, \infty)$, 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:

$$\text{OPT}(\mathcal{I}_s^{>i}) := \max_{j \in \{i+1, \ldots, m\}} (p_{e_j} \cdot w_{e_j} + (1 - p_{e_j}) \cdot \text{OPT}(\mathcal{I}_s^{>j})) \tag{2.2}$$

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 (2.2) to recover an optimal solution to f. We can define DP-OPT to be a memoization based implementation of (2.2). It is clear DP-OPT can be implemented in the claimed time complexity. ◀

Given $R \subseteq U$, consider the induced stochastic graph, $G[\{s\} \cup R]$ for $R \subseteq U$ which has probing constraint $\mathcal{C}_s^R \subseteq \mathcal{C}_v$, constructed by restricting $\mathcal{C}_s$ to those strings whose entries all lie in $R \times \{s\}$. Moreover, 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 1 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 $G = (U, V, E)$.

```
M ← ∅.
R ← U.
for t = 1, ..., n do
    Let v_t be the current online arrival node, with constraint C_{v_t}.
    Set e ← DP-OPT(v_t, R)
    for i = 1, ..., |e| do
        Probe e_i.
        if st(e_i) = 1 then
            Add e_i to M, and update R ← R \ {u_i}, where e_i = (u_i, v_t).
return M.
```

In general, the behaviour of the committal benchmark, namely $\text{OPT}(s, R)$, can change very much, even for minor changes to R. For instance, if $R = U$, then $\text{OPT}(s, U)$ may probe the edge (u, s) first – thus giving it highest priority – whereas if $u^* \in U$ is removed from U (where $u^* \neq u$), $\text{OPT}(s, U \setminus \{u^*\})$ may not probe (u, v) at all (see Example B.1 for an explicit instance of this behaviour). As a result, it is easy to consider an execution of Algorithm 1 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 1 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 1.1 because our primal-dual charging assignment does not depend on the order of the online vertex arrivals (see Appendix B). This contrasts with the ROM setting, in which Example B.1 can be extended to show that the cost sharing rule $g(z) := \exp(z - 1)$ will not work in general. Our approach is thus to restrict our attention to stochastic graphs in which executions of Algorithm 1 are either monotonic, or monotonic with high probability. This leads us to the definition of rankability, which characterizes a large number of settings in which Algorithm 1 is monotonic.

Given a vertex $v \in V$, and an ordering π_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 π_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 π_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 π_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 π_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 $\text{DP-OPT}(v, R)$ and $\pi_v(R)$ are equal. Crucially, if v is rankable, then when vertex v arrives while executing Algorithm 1, one can compute the ranking π_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 $\text{OPT}(v, R)$. We consider three (non-exhaustive) examples of rankability:

▶ **Proposition 2.2.** *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 ℓ_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 $b_{u_1,v} \geq b_{u_2,v}$.*

▶ Remark 2.3. Note that the cases of Proposition 2.2 subsume all the settings listed in the abstract. 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 vertices are themselves rankable. We emphasize that distinct vertices of V may each use their own separate rankings of their adjacent edges.

As discussed in Subsection 1.3, the 0.544 inapproximation against LP-std [8] prevents us from proving a performance guarantee against LP-std, even for patience values. We instead upper bound $\text{OPT}(G)$ using a tighter LP relaxation that comes with the additional benefit of applying to downward-closed probing constraints. For each $u \in U$ and $v \in V$, let $x_{u,v}$ be a decision variable corresponding to the probability that $\text{OPT}(G)$ probes the edge (u, v).

$$\begin{aligned}
\text{maximize} \quad & \sum_{u \in U} \sum_{v \in V} w_u \cdot p_{u,v} \cdot x_{u,v} && && \text{(LP-DP)} \\
\text{subject to} \quad & \sum_{v \in V} p_{u,v} \cdot x_{u,v} \leq 1 && \forall u \in U && (2.3) \\
& \sum_{u \in R} w_u \cdot p_{u,v} \cdot x_{u,v} \leq \text{OPT}(v, R) && \forall v \in V,\ R \subseteq U && (2.4) \\
& x_{u,v} \geq 0 && \forall u \in U, v \in V && (2.5)
\end{aligned}$$

Denote $\text{LPOPT}_{\text{DP}}(G)$ as the optimal value of this LP. Constraint (2.3) can be viewed as ensuring that the expected number of matches made to $u \in U$ is at most 1. Similarly, (2.4) 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 committal benchmark. Thus, $\text{OPT}(G) \leq \text{LPOPT}_{\text{DP}}(G)$ (a formal proof specific to patience values is proven in [8]).

2.0.1 Defining the Primal-Dual Charging Schemes

In order to prove Theorems 1.1 and 1.2, we employ primal-dual charging arguments based on the dual of LP-DP. For each $u \in U$, define the variable α_u. Moreover, for each $R \subseteq U$ and $v \in V$, define the variable $\phi_{v,R}$ (these latter variables correspond to constraint (2.4)).

$$\begin{aligned}
\text{minimize} \quad & \sum_{u \in U} \alpha_u + \sum_{v \in V} \sum_{R \subseteq U} \text{OPT}(v, R) \cdot \phi_{v,R} && && \text{(LP-dual-DP)} \\
\text{subject to} \quad & p_{u,v} \cdot \alpha_u + \sum_{\substack{R \subseteq U: \\ u \in R}} w_u \cdot p_{u,v} \cdot \phi_{v,R} \geq w_u \cdot p_{u,v} && \forall u \in U, v \in V && (2.6) \\
& \alpha_u \geq 0 && \forall u \in U && (2.7) \\
& \phi_{v,R} \geq 0 && \forall v \in V, R \subseteq U && (2.8)
\end{aligned}$$

The dual-fitting argument used to prove Theorem 1.2 has an initial set-up which proceeds similarly to the argument in Devanur et al. [10]. Specifically, 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/cost sharing. 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 $F = F(G)$, where

$$F(G) := \begin{cases} 1 - \frac{1}{e} & G \text{ is rankable} \\ \left(1 - \frac{1}{e}\right) \cdot \min_{v \in V} (1 - p_v)^{c_v} & \text{otherwise} \end{cases} \tag{2.9}$$

In order to prove Theorem 1.2, we shall prove that Algorithm 1 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 setting. 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 1 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 1 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 1 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 α_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,

$$\mathbb{E}[w(\mathcal{M})] = F \cdot \left(\sum_{u \in U} \mathbb{E}[\alpha_u] + \sum_{v \in V} \sum_{R \subseteq U} \text{OPT}(v, R) \cdot \mathbb{E}[\phi_{v,R}] \right), \tag{2.10}$$

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 (2.10) implies the following lemma:

▶ **Lemma 2.4.** *Suppose $G = (U, V, E)$ is a stochastic graph for which Algorithm 1 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 2.5.** *If the online nodes of $G = (U, V, E)$ are presented to Algorithm 1 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 committal benchmark, Theorem 1.2 follows from Lemmas 2.4 and 2.5 in conjunction with weak duality.

2.0.2 Proving Dual Feasibility: Lemma 2.5

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 2.0.1. In order to prove Lemma 2.5, we must show that for each fixed $u_0 \in U$ and $v_0 \in V$, we have that

$$\mathbb{E}[p_{u_0,v_0} \cdot \alpha_{u_0} + w_{u_0} \cdot p_{u_0,v_0} \sum_{\substack{R \subseteq U: \\ u_0 \in R}} \phi_{v_0,R}] \geq w_{u_0} \cdot p_{u_0,v_0}. \tag{2.11}$$

Our strategy for proving (2.11) first involves the same approach as used in Devanur et al. [10]. 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 α_{u_0} in terms of the critical time $\widetilde{Y}_c$.

▶ **Proposition 2.6.**
- *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 2.7. Note that Proposition 2.6 is the only part of the proof of Theorem 1.2 which is affected by whether or not G is rankable. We defer the proof of Proposition 2.6 to Appendix B.

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 2.8.**

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

Proof of Proposition 2.8. 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 1 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

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

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 the values of $(Y_v)_{v \in V}$, as well as the states of the edges of $\widetilde{E}$; that is, $(\mathrm{st}(e))_{e \in \widetilde{E}}$. Observe that once we condition on this information, we can determine $g(Y_{v_0})$, as well as R_{v_0}. As such,

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

On the other hand, the only randomness which remains in the conditional expectation involving $w(\mathcal{M}(v_0))$ is over the states of the edges adjacent to v_0. Observe now that since Algorithm 1 behaves optimally on $G[\{v_0\} \cup R_{v_0}]$, we get that

$$\mathbb{E}[w(\mathcal{M}(v_0)) \mid (Y_v)_{v \in V}, (\mathrm{st}(e))_{e \in \widetilde{E}}] = \mathrm{OPT}(v_0, R_{v_0}), \tag{2.12}$$

and so for the *fixed* subset $R \subseteq U$,

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

after multiplying each side of (2.12) by the indicator random variable $\mathbf{1}_{[R_{v_0} = R]}$. Thus,

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

after cancellation. We therefore get that

$$\sum_{\substack{R \subseteq U: \\ u_0 \in R}} \mathbb{E}[\phi_{v_0,R} \,|\, (Y_v)_{v \in V}, (\mathrm{st}(e))_{e \in \widetilde{E}}] = \frac{g(Y_{v_0})}{F} \sum_{\substack{R \subseteq U: \\ u_0 \in R}} \mathbf{1}_{[R_{v_0} = R]}.$$

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,

$$\sum_{\substack{R \subseteq U: \\ u_0 \in R}} \mathbb{E}[\phi_{v_0,R} \,|\, (Y_v)_{v \in V}, (\mathrm{st}(e))_{e \in \widetilde{E}}] \geq \frac{g(Y_{v_0})}{F} \mathbf{1}_{[Y_{v_0} < \widetilde{Y}_c]}.$$

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

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

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,

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

and so the claim holds.

◀

With Propositions 2.6 and 2.8, the proof of Lemma 2.5 follows easily (see Appendix B), and so Theorem 1.2 is proven.

3 Edge Weights

Let us suppose that $G = (U, V, E)$ is a stochastic graph with arbitrary edge weights, probabilities and downward-closed probing constraints $(\mathcal{C}_v)_{v \in V}$. For each $k \geq 1$ and $\boldsymbol{e} = (e_1, \ldots, e_k) \in E^{(k)}$, define $g(\boldsymbol{e}) := \prod_{i=1}^{k} (1 - p_{e_i})$. Notice that $g(\boldsymbol{e})$ corresponds to the probability that all the edges of $\boldsymbol{e}$ are inactive, where $g(\lambda) := 1$ for the empty string λ. We also define $\boldsymbol{e}_{<e_i} := (e_1, \ldots, e_{i-1})$ for each $2 \leq i \leq k$, which we denote by $\boldsymbol{e}_{<i}$ when clear. By convention, $\boldsymbol{e}_{<1} := \lambda$. Observe then that $\mathrm{val}(\boldsymbol{e}) := \sum_{i=1}^{|\boldsymbol{e}|} p_{e_i} w_{e_i} \cdot g(\boldsymbol{e}_{<i})$ corresponds to the expected weight of the first active edge if $\boldsymbol{e}$ is probed in order of its indices, where $\mathrm{val}(\lambda) := 0$.

For each $v \in V$, we introduce a decision variable denoted $x_v(\boldsymbol{e})$, which may loosely be interpreted as the likelihood the committal benchmark probes the edges in the order specified by $\boldsymbol{e} = (e_1, \ldots, e_k)$ [6]. With this notation, we express the following LP:

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

$$\text{subject to} \quad \sum_{v \in V} \sum_{\substack{\boldsymbol{e} \in \mathcal{C}_v: \\ (u,v) \in \boldsymbol{e}}} p_{u,v} \cdot g(\boldsymbol{e}_{<(u,v)}) \cdot x_v(\boldsymbol{e}) \leq 1 \qquad \forall u \in U \tag{3.1}$$

$$\sum_{\boldsymbol{e} \in \mathcal{C}_v} x_v(\boldsymbol{e}) = 1 \qquad \forall v \in V, \tag{3.2}$$

$$x_v(\boldsymbol{e}) \geq 0 \qquad \forall v \in V, \boldsymbol{e} \in \mathcal{C}_v \tag{3.3}$$

Denote $\text{LPOPT}_{\text{conf}}(G)$ as the optimal value of LP-config. This LP was developed from insights relevant to both the secretary and prophet settings. Specifically, the DP-OPT algorithm of Theorem 2.1 can be used as a (deterministic) polynomial time separation oracle for the dual of LP-config. This ensures that LP-config can be solved in polynomial time as a consequence of how the ellipsoid algorithm [26, 11] executes (see Theorem A.1 in Appendix A for details). In [5], we prove that LP-config is a relaxation of the committal benchmark. Unlike previous LP relaxations of the committal benchmark, we are not aware of an easy proof of this fact, and we consider it to be a technical contribution.

We now define a *fixed vertex* probing algorithm, called VertexProbe, which is applied to an online vertex s of an arbitrary stochastic graph (potentially distinct from G) with probing constraints $\mathcal{C}_s$ on $\partial(s)$. Specifically, given non-negative values $(z(\boldsymbol{e}))_{\boldsymbol{e} \in \mathcal{C}_s}$ which satisfy $\sum_{\boldsymbol{e} \in \mathcal{C}_s} z(\boldsymbol{e}) = 1$, draw $\boldsymbol{e}'$ with probability $z(\boldsymbol{e}')$. If $\boldsymbol{e}' = (e'_1, \ldots, e'_k)$ for $k := |\boldsymbol{e}'| \geq 1$, then probe the edges of $\boldsymbol{e}'$ in order, and match s to the first edge revealed to be active. If no such edge exists, or $\boldsymbol{e}' = \lambda$, then return $\emptyset$.

▶ **Lemma 3.1.** *Suppose* VertexProbe *is passed a fixed online node s of a stochastic graph, and values $(z(\boldsymbol{e}))_{e \in \mathcal{C}_s}$ which satisfy $\sum_{\boldsymbol{e} \in \mathcal{C}_s} z(\boldsymbol{e}) = 1$. If for each $e \in \partial(s)$,*

$$\widetilde{z}_e := \sum_{\substack{\boldsymbol{e}' \in \mathcal{C}_v: \\ e \in \boldsymbol{e}'}} g(\boldsymbol{e}'_{<e}) \cdot z_v(\boldsymbol{e}'),$$

then e is probed with probability $\widetilde{z}_e$, and returned by the algorithm with probability $p_e \cdot \widetilde{z}_e$.

▶ Remark 3.2. If VertexProbe outputs the edge $e = (u, s)$ when executing on the fixed node s, then we say that s **commits** to the edge $e = (u, s)$, or that s commits to u.

Returning to the problem of designing an online probing algorithm for G, let us assume that $n := |V|$, and that the online nodes of V are denoted $v_1, \ldots, v_n$, where the order is generated *u.a.r.* Denote V_t as the set of first t arrivals of V; that is, $V_t := \{v_1, \ldots, v_t\}$. Moreover, set $G_t := G[U \cup V_t]$, and $\text{LPOPT}_{\text{conf}}(G_t)$ 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 3.3.** *For each $t \geq 1$, $\mathbb{E}[LPOPT_{conf}(G_t)] \geq \frac{t}{n} LPOPT_{conf}(G)$.*

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,

[6] While this is the natural interpretation of the decision variables of LP-config, to the best of our knowledge, formally defining the variables in this way does not lead to a proof that LP-config relaxes the committal benchmark. We discuss this in detail in [5].

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 2 Unknown Stochastic Graph ROM

```
Input: U and n := |V|.
Output: a matching M from the (unknown) stochastic graph G = (U, V, E) of active edges.
 1: Set M ← ∅.
 2: Set G_0 = (U, ∅, ∅)
 3: for t = 1, ..., n do
 4:     Input v_t, with (w_e)_{e∈∂(v_t)}, (p_e)_{e∈∂(v_t)} and C_{v_t}.
 5:     Compute G_t, by updating G_{t-1} to contain v_t (and its relevant information).
 6:     if t < ⌊n/e⌋ then
 7:         Pass on v_t.
 8:     else
 9:         Solve LP-config for G_t and find an optimal solution (x_v(e))_{v∈V_t, e∈C_v}.
10:         Set e_t ← VertexProbe(v_t, ∂(v_t), (x_v(e))_{e∈C_{v_t}}).
11:         if e_t = (u_t, v_t) ≠ ∅ and u_t is unmatched then
12:             Add e_t to M.
13: return M.
```

▶ Remark 3.4. Unlike the algorithm of Kesselheim et al., our algorithm is randomized, and we do not know whether the polytope LP-config always admits an optimum integral solution. We leave it as an interesting open question as to whether or not Algorithm 2 can be derandomized.

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 $t \geq \lceil \alpha n \rceil$. For each $\alpha n \leq t \leq n$, define R_t as the *unmatched vertices* of U when vertex v_t arrives. Note that committing to $e_t = (u_t, v_t)$ is necessary, but not sufficient, for v_t to match to u_t. 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 3.5.** *For each $t \geq \lceil \alpha n \rceil$, $\mathbb{E}[w(e_t)] \geq LPOPT_{conf}(G)/n$.*

▶ **Lemma 3.6.** *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* [7].

The proofs of Lemmas 3.5 and 3.6 mostly follow the analogous claims as proven by Kesselheim et al in the classic secretary matching problem. We present formal proofs in the arXiv version [4]. With these lemmas, together with the efficient solvability of LP-config, the proof of Theorem 1.6 follows easily (see Appendix C).

4 Conclusion and Open Problems

We considered the online stochastic bipartite matching with commitment in a number of different settings establishing several competitive bounds against the committal benchmark. Our work leaves open a number of challenging problems. For context we note that currently, even for the classical (i.e., non-probing) setting, $1 - \frac{1}{e}$ is the best known ratio for deterministic

[7] 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.

algorithms operating on unweighted or vertex weighted graphs with random order vertex arrivals. The best known ROM inapproximation of 0.823 (due to Manshadi et al. [21]) comes from the classical i.i.d. unweighted graph setting for a known distribution and applies to randomized as well as deterministic algorithms.

- What is the best ratio that a deterministic or randomized online algorithm can obtain for *all* vertex weighted stochastic graphs in the ROM setting? That is, what competitive ratio can be achieved without the rankable assumption? Is there an online probing algorithm which can surpass the $1 - 1/e$ "barrier" with or without the rankable assumption? Here we note that in the classical ROM setting, the Ranking algorithm achieves a 0.696 ratio for unweighted graphs (due to Mahdian and Yan [20]) and a 0.6534 ratio (due to Huang et al. [15]) for vertex weighted graphs. Thus, randomization seems to significantly help in the classical ROM setting.
- What is the best ratio that a randomized online algorithm can obtain for stochastic graphs in the adversarial arrival model? The Mehta and Panigraphi [22] 0.621 inapproximation shows that randomized probing algorithms (even for unweighted graphs and unit patience) cannot achieve a $1 - 1/e$ performance guarantee against LP-std-unit, however the work of Goyal and Udwani [12] suggests that this is because LP-std-unit is too loose a relaxation of the committal benchmark.
- For edge weighted graphs, can we achieve a $\frac{1}{e}$ competitive ratio (or any constant ratio) by a combinatorial (and more efficient) algorithm? Our vertex weighted algorithm can be viewed as a truthful online (or random order) posted price mechanism. Can we modify the edge weighted algorithm to be a truthful mechanism thereby generalizing the truthful mechanism of Reiffenhauser [25]? Note that unlike the vertex weighted algorithm, our algorithm for edge weights does not necessarily make an optimal social welfare decision for each online node.

Acknowledgement

We would like to thank Denis Pankratov, Rajan Udwani, and David Wajc for their very constructive comments on 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. *CoRR*, abs/1505.01439, 2015.

4 Allan Borodin, Calum MacRury, and Akash Rakheja. Greedy approaches to online stochastic matching. *CoRR*, abs/2008.09260, 2020.

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

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

7 Brian Brubach, Nathaniel Grammel, Will Ma, and Aravind Srinivasan. Improved guarantees for offline stochastic matching via new ordered contention resolution schemes. *CoRR*, abs/2106.06892, 2021.

8 Brian Brubach, Nathaniel Grammel, and Aravind Srinivasan. Vertex-weighted online stochastic matching with patience constraints. *2019*, 1907.03963, 2019.

9 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.

10 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.

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

12 Vineet Goyal and R. Udwani. Online matching with stochastic rewards: Optimal competitive ratio via path based formulation. *Proceedings of the 21st ACM Conference on Economics and Computation*, 2020.

13 Nathaniel Grammel, Brian Brubach, Will Ma, and Aravind Srinivasan. Follow your star: New frameworks for online stochastic matching with known and unknown patience. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, pages 2872–2880, 2021.

14 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.

15 Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals, 2018.

16 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.

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

18 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.

19 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.

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

21 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.

22 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.

23 Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. In *SODA*, pages 1388–1404, 2015.

24 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 June 2019.

25 Rebecca Reiffenhäuser. An optimal truthful mechanism for the online weighted bipartite matching problem. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1982–1993. SIAM, 2019.

26 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.

27 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.

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

A Solving LP-config Efficiently

Suppose that we are given an arbitrary stochastic graph $G = (U, V, E)$. We contrast LP-config with LP-std, which is defined only when G has patience values $(\ell_v)_{v \in V}$:

$$\begin{aligned}
&\text{maximize} && \sum_{e \in E} w_e \cdot p_e \cdot x_e && && \text{(LP-std)} \\
&\text{subject to} && \sum_{e \partial(u)} p_e \cdot x_e \leq 1 && \forall u \in U && \text{(A.1)} \\
& && \sum_{e \in \partial(v)} p_e \cdot x_e \leq 1 && \forall v \in V && \text{(A.2)} \\
& && \sum_{e \in \partial(v)} x_e \leq \ell_v && \forall v \in V && \text{(A.3)} \\
& && 0 \leq x_e \leq 1 && \forall e \in E. && \text{(A.4)}
\end{aligned}$$

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

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

A.1 Solving LP-config Efficiently

We now show how LP-config be solved efficiently under the assumptions of Theorem 1.6.

▶ **Theorem A.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 is efficiently solvable in $|G|$.*

We prove Theorem A.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.

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

Observe that to prove Theorem A.1, it suffices to show that LP-config-dual has a (deterministic) polynomial time separation oracle, as a consequence of how the ellipsoid algorithm [26, 11] executes (see [28, 27, 2, 19] 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 $\boldsymbol{e} = (e_1, \ldots, e_{|\boldsymbol{e}|}) \in \mathcal{C}_v$, such that if $e_i = (u_i, v)$ for $i = 1, \ldots, k$, then

$$f(\boldsymbol{e}) := \sum_{j=1}^{|\boldsymbol{e}|} (w_{e_j} - \alpha_{u_j}) \cdot p_{e_j} \cdot g(\boldsymbol{e}_{<j}) > \beta_v, \tag{A.8}$$

where $f(\boldsymbol{e}) := 0$ if $\boldsymbol{e} = \lambda$.

▶ **Lemma A.2.** *In the membership oracle model,* DP-OPT *of Proposition 2.1 can be used to efficiently check whether $f(\boldsymbol{e}') > \beta_v$ for some $\boldsymbol{e}' \in \mathcal{C}_v$, provided $\mathcal{C}_v$ is downward-closed. Moreover, if such a tuple exists, then it can be found efficiently.*

Proof. In order to make this statement, it suffices to show how one can use DP-OPT to maximize the function f efficiently.

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 (A.8) is maximized by the empty-string λ. 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 maximizing f corresponds to executing DP-OPT on the stochastic graph $G[U \cup \{v\}]$, with edge weights replaced by $(\widetilde{w}_e)_{e \in \partial(v)}$.

◀

B Proofs and Additions to Section 2

Proof of Theorem 1.1. Let $G = (U, V, E)$ be a vertex weighted stochastic graph, and assume that Algorithm 1 returns the matching $\mathcal{M}$ when the online vertices of G are presented to the algorithm in adversarial order.

We now define a charging assignment as Algorithm 1 executes on G. First, 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. Let us now take $v \in V, u \in U$, and $R \subseteq U$, where $u \in R$. If R consists of the unmatched vertices when v it arrives, then suppose that Algorithm 1 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 α_u and $w_u/\mathrm{OPT}(v, R)$ to $\phi_{v,R}$. Observe that each subset $R \subseteq U$ is charged at most once, as is each $u \in U$. Thus,

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

where the expectation is over $(\mathrm{st}(e))_{e \in E}$. 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 claim that $((\alpha_u^*)_{u \in U}, (\phi_{v,R}^*)_{v \in V, R \subseteq U})$ is a feasible solution to LP-dual-DP. To show this, we must prove that for each fixed $u_0 \in U$ and $v_0 \in V$, we have that

$$\mathbb{E}[p_{u_0,v_0} \cdot \alpha_{u_0} + w_{u_0} \cdot p_{u_0,v_0} \sum_{\substack{R \subseteq U: \\ u_0 \in R}} \phi_{v_0,R}] \geq w_{u_0} \cdot p_{u_0,v_0}. \tag{B.2}$$

We first define R_{v_0} as 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 inequality:

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

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,

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

On the other hand, the only randomness which remains in the conditional expectation involving $w(\mathcal{M}(v_0))$ is over $(\mathrm{st}(e))_{e \in \partial(v_0)}$. However, since Algorithm 1 behaves optimally on $G[\{v_0\} \cup R_{v_0}]$, we get that

$$\mathbb{E}[w(\mathcal{M}(v_0)) \,|\, (Y_v)_{v \in V}, (\mathrm{st}(e))_{e \in \widetilde{E}}] = \mathrm{OPT}(v_0, R_{v_0}), \tag{B.3}$$

and so for the *fixed* subset $R \subseteq U$,

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

after multiplying each side of (B.3) by the indicator random variable $\mathbf{1}_{[R_{v_0} = R]}$. Thus,

$$\mathbb{E}[\phi_{v_0,R} \,|\, (\mathrm{st}(e))_{e \in \widetilde{E}}] = \mathbf{1}_{[R_{v_0} = R]},$$

after cancellation. We therefore get that

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

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 α_{u_0} must have been charged w_u. In other words, $\alpha_{u_0} \geq w_u \cdot \mathbf{1}_{[u_0 \notin R_{v_0}]}$. As a result,

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

and so (B.2) follows after taking expectations. The solution $((\alpha^*_u)_{u \in U}, (\phi^*_{v,R})_{v \in V, R \subseteq U})$ is therefore feasible, and so since $\mathrm{OPT}(G) \leq \mathrm{LPOPT}_{\mathrm{DP}}(G)$, the proof is complete after applying weak duality and (B.1). ◀

▶ **Example B.1.** 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 ε is sufficiently small – concretely, $\varepsilon \leq 1/12$. If $R_1 := U$, then $\mathrm{OPT}(v, R_1)$ probes (u_1, v) and then (u_2, v) in order. On the other hand, if $R_2 = R_1 \setminus \{v_2\}$, then $\mathrm{OPT}(v, R_2)$ does *not* probe (u_1, v). Specifically, $\mathrm{OPT}(v, R_2)$ probes (u_3, v) and then (u_4, v).

Proof of Proposition 2.6. For each $v \in V$, denote $R^{\mathrm{af}}_v(G)$ 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 $R^{\mathrm{af}}_v(G)$. Similarly, define $R^{\mathrm{af}}_v(\widetilde{G})$ 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

$$R^{\mathrm{af}}_v(G) \subseteq R^{\mathrm{af}}_v(\widetilde{G}), \tag{B.4}$$

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 $u_0 \notin R^{\mathrm{af}}_{v_c}(\widetilde{G})$ and thus by (B.4), that $u_0 \notin R^{\mathrm{af}}_{v_c}(G)$. 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,

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

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 $(\mathrm{st}(e))_{e \in \widetilde{E}}$ and $(Y_v)_{v \in V}$, and recall the definitions of $p_{v_0} := \max_{e \partial(v_0)} p_e$ and $c_{v_0} := \max_{\boldsymbol{e} \in \mathcal{C}_{v_0}} |\boldsymbol{e}|$. Observe that if every probe involving an edge of

[8] Example B.1 shows why (B.4) will not hold if G is not rankable.

$\partial(v_0)$ 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,

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

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

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

and so the proof is complete. ◀

Proof of Lemma 2.5. We first observe that by taking the appropriate conditional expectation, Proposition 2.6 ensures that

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

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 2.8,

$$\mathbb{E}[p_{u_0,v_0} \cdot \alpha_{u_0} + w_{u_0} \cdot p_{u_0,v_0} \cdot \sum_{\substack{R \subseteq U: \\ u_0 \in R}} \phi_{v,R} \,|\, (Y_v)_{v \in \widetilde{V}}, (\mathrm{st}(e))_{e \in \widetilde{E}}], \tag{B.5}$$

is lower bounded by

$$\left(1 - \frac{1}{e}\right)^{-1} w_{u_0} \cdot p_{u_0,v_0} \cdot (1 - g(\widetilde{Y}_c)) + \frac{w_{u_0}\, p_{u_0,v_0}}{F} \int_0^{\widetilde{Y}_c} g(z)\, dz. \tag{B.6}$$

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,

$$\left(1 - \frac{1}{e}\right)^{-1} \left((1 - g(\widetilde{Y}_c)) + \frac{1 - 1/e}{F} \int_0^{\widetilde{Y}_c} g(z)\, dz\right) \geq 1, \tag{B.7}$$

as $F \leq 1 - 1/e$ by definition (see (2.9)). If we now lower bound (B.6) using (B.7) and take expectations over (B.5), 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 2.5 is feasible, and so the proof is complete.

◀

C Proofs and Additions to Section 3

Proof of Theorem 1.6. Clearly, Algorithm 2 can be implemented efficiently, since LP-config is efficiently solvable. Thus, we focus on proving the algorithm attains the desired asymptotic competitive ratio.

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 as the edge v_t commits to, which is the empty-set by definition if no such commitment is made. Observe that

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

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 3.6 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[9]

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

As a result, using (C.1), we get that

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

We may thus conclude that

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

after applying Lemma 3.5. As $\sum_{t=\lceil \alpha n \rceil}^{n} f(t, n)/n \geq (1/e - 1/n)$, the result holds. ◀

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