Follow Your Star: New Frameworks for Online Stochastic Matching with Known and Unknown Patience

Brian Brubach
bb100@wellesley.edu

Nathaniel Grammel
ngrammel@umd.edu

Will Ma
wm2428@gsb.columbia.edu

Aravind Srinivasan
asriniv1@umd.edu

Abstract

We study several generalizations of the Online Bipartite Matching problem. We consider settings with stochastic rewards, patience constraints, and weights (considering both vertex- and edge-weighted variants). We introduce a *stochastic* variant of the patience-constrained problem, where the patience is chosen randomly according to some known distribution and is not known in advance. We also consider *stochastic* arrival settings (i.e. the nature in which the online vertices arrive is determined by a known random process), which are natural settings that are able to beat the hard worst-case bounds of adversarial arrivals.

We design black-box algorithms for star graphs under various models of patience, which solve the problem optimally for deterministic or geometrically-distributed patience, and yield a 1/2-approximation for any patience distribution. These star graph algorithms are then used as black boxes to solve the online matching problems under different arrival settings. We show improved (or first-known) competitive ratios for these problems. We also present negative results that include formalizing the concept of a stochasticity gap for LP upper bounds on these problems, showing some new stochasticity gaps for popular LPs, and bounding the worst-case performance of some greedy approaches.

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

1 Introduction

Online matching is a fundamental problem in e-commerce and online advertising, introduced in the seminal work of Karp et al. (1990). While offline matching has a long history in economics and computer science, online matching has exploded in popularity with the ubiquity of the internet and emergence of online marketplaces. A common scenario in e-commerce is the online sale of unique goods due to the ability to reach niche markets via the internet (e.g., eBay). We will use this as a motivating example to describe our setting. However, the settings we study can also model job search/hiring, crowdsourcing, online advertising, ride-sharing, and other online-matching problems.

In classical online bipartite matching, we start with a known set of *offline vertices* U that may represent items for sale or ads to be allocated. Then, there is an unknown set V of *online vertices*, which may represent customers, users, or visitors to a webpage. The vertices of V arrive online in some fashion. Typically, the customers arrive one-by-one, and a decision to match each customer or not (and if so, to which item) must be made irrevocably before the next customer is revealed. In the original formulation, these online vertices arrived in adversarial order (Karp et al., 1990). However, less pessimistic *stochastic* arrival models were introduced to better model real-world applications (e.g. e-commerce). These models have been studied extensively in subsequent work and include random order arrival, known independent and identically distributed arrivals (known IID), and prophet inequality. The present work addresses all of these models (adversarial and random) either directly or through known extensions.

In addition to a variety of arrival models, there are generalizations for *weighted* graphs and *stochastic rewards*. *Vertex weights* on the offline vertices correspond to item prices/profits or the reward for making a match. More general *edge weights* allow some items to be offered at different prices to different customers. In the *stochastic rewards* model (also called *stochastic matching*[1]), each edge has a known and independent probability of existing (Bansal et al., 2010; Mehta and Panigrahi, 2012). When an online vertex arrives, the probabilities of each of its incident edges are revealed. These probabilities may correspond to the probability that a customer buys an offered item or that a user clicks an ad (in *pay-per-click* advertising). See Mehta (2012) for further discussion.

In the e-commerce and advertising settings, we discover if a customer or user would purchase an item or click an ad *after* we have presented it to them and they have done so (or not). We cannot later choose to "revoke" the item offer or ad placement. This is the *probe-commit* model: if a stochastic edge is probed and exists, it must be matched irrevocably.

In the basic *stochastic rewards* setting, we may probe at most one edge incident to an online vertex before it becomes unavailable for further match attempts; there is no limit to the number of edges probed incident to any offline vertex (Mehta and Panigrahi, 2012; Brubach et al., 2016). Think of a single banner ad on a website for example. A generalization of this setting is the introduction of *patience constraints* (also known as *timeouts* in the literature) where an online vertex v has a known patience θ_v, and we may probe up to θ_v neighbors, stopping early if v is successfully matched (Bansal et al., 2010; Adamczyk et al., 2015; Brubach et al., 2017). This models many natural situations, where users and customers may be presented with multiple items or ads in sequence. The classic stochastic rewards problem without patience constraints (Mehta and Panigrahi, 2012) corresponds to the special case $\theta_v = 1$ for all v. As in that variant, we assume that all offline vertices have unlimited patience. This is a standard assumption (see, e.g., Bansal et al. (2010)), although two-sided patience has been studied (Brubach et al., 2017).

Finally, we introduce a further variant of the patience-constrained model: stochastic, unknown patience values. To our knowledge, this is the first time such models have been studied in the matching literature. Essentially, the patience θ_v of each online vertex is drawn randomly from some known distribution, but is not revealed to the algorithm until it has unsuccessfully probed θ_v edges, at which point (as with the known patience setting) the online vertex becomes unavailable for matching and no reward is obtained. We introduce two models of uncertain patience: the "constant hazard rate" model (each online vertex has a fixed known probability of exhausting its patience after a failed probe) and the arbitrary patience model (where the probability distribution on the patience is given explicitly).

[1]The term "online stochastic matching" is sometimes used to describe online-matching problems with stochastic arrival models (e.g., known IID arrivals) in the literature. We use this term to refer to matching with stochastic edges as in the offline stochastic matching problem.

1.1 Related Work

Online matching was introduced in Karp et al. (1990) for unweighted graphs with adversarial arrivals. Since then, a vast landscape of variants have been studied, and we mention the most relevant here. The online arrivals can be adversarial, random order, known IID, or known non-identical (also called prophet-inequality matching). The graph can be unweighted, vertex-weighted (on the offline vertices aka posted prices), or edge-weighted. The edges can be deterministic (aka classical matching) or stochastic. Online vertices with stochastic edges can have a known, arbitrary patience value or all online vertices can have a patience of one (the stochastic-rewards variant). We do not include unknown patience in this section since we are not aware of prior work on the unknown-patience models we introduce.

1.1.1 Deterministic Edges

For the *adversarial* arrivals setting, the classic Ranking algorithm achieves a tight competitive ratio of $1 - 1/\mathsf{e}$ on unweighted graphs (Karp et al., 1990), while a modification yields $1 - 1/\mathsf{e}$ (Aggarwal et al., 2011) on vertex-weighted graphs. However, the edge-weighted problem can have an arbitrarily bad competitive ratio in this setting without additional allowances such as free disposal (Feldman et al., 2009; Fahrbach et al., 2020) or weight-dependent competitive ratios (Ma and Simchi-Levi, 2020).

In *random order* arrivals, the unknown bipartite graph is constructed adversarially, but online vertices arrive in a uniform random permutation. The best known competitive ratios for deterministic edges are 0.696 (Mahdian and Yan, 2011), 0.6534 (Huang et al., 2018), and $1/\mathsf{e}$ (Kesselheim et al., 2013) for unweighted, vertex-weighted, and edge-weighted graphs, respectively. There is a hardness of $5/6$ (Goel and Mehta, 2008).

In the *known IID* setting, we are given a bipartite graph upfront with the online partition representing vertex "types." Each arrival is sampled with replacement from a known distribution over these online vertices. Multiple copies of the same online vertex from this original graph may arrive over time and are treated as distinct vertices. Thus, the online vertex that arrives at each time step is independent and identically distributed (IID) with respect to all other online vertices. Here, the current best results are 0.7299 (Brubach et al., 2016) for vertex-weighted (and by extension, unweighted) and 0.705 (Brubach et al., 2016) for edge-weighted graphs.

The *known non-identical* problem, also called *prophet inequality* matching (Alaei et al., 2012), may be viewed as a variant of edge-weighted b-matching (offline vertices can be matched up to b times) with an arrival model that generalizes known IID. At each time step the online arrival may have a *different* (though still independent and known) distribution. For the classical matching case ($b = 1$), Alaei et al. (2012) achieve a competitive ratio of 0.5.

1.1.2 Online Stochastic Matching

Online stochastic matching was introduced in Bansal et al. (2010) as stochastic matching with timeouts (patience). They considered known IID arrivals and showed a ratio of 0.12 which was eventually improved to 0.46 (Brubach et al., 2017)[2]. We refer to this model simply as stochastic matching with patience. The special case of $\theta_v = 1$ for all v was later studied by Mehta and Panigrahi (2012) under adversarial arrivals. Under the restricted case of uniform edge probabilities, they showed that 0.53 is possible. This was extended to a ratio of 0.534 for unequal, but vanishingly small probabilities (Mehta et al., 2015). However, for arbitrary edge probabilities, a trivial 0.5 ratio is the best known. There is also a hardness result in Mehta and Panigrahi (2012) which claims that no algorithm for stochastic rewards with adversarial arrivals can achieve a competitive ratio greater than 0.62 (strictly less than $1 - 1/\mathsf{e}$). We note that this hardness result is not applicable to our work since it uses a different, more pessimistic definition of competitive ratio which compares the online algorithm to a linear program rather than the offline stochastic matching problem.

[2]The techniques in Brubach et al. (2017) also involved solving a star graph problem with a black box. However, that work first solved a linear program for a bipartite graph, then used a black probing algorithm to essentially round and probe the LP solution on the induced star graphs of arriving vertices. This differs from our work which uses algorithms for stochastic matching on star graphs as black boxes to solve a more sophisticated LP, then use that LP solution to guide the online algorithm.

The very recent work of Borodin et al. (2020) gives a competitive ratio of $1 - 1/\mathsf{e}$ under random order arrivals for three distinct cases: vertex-weighted stochastic rewards ($\theta_v = 1$), unweighted with arbitrary patience, and vertex-weighted with arbitrary patience, but with all edges incident on an online vertex having the same probability. The work of Borodin et al. (2020) also independently and concurrently achieves the same $1 - 1/\mathsf{e}$ ratio for the known IID setting with edge weights as the present work, using similar techniques.

Another closely related model is that of Goyal and Udwani (2020), which includes stochastic rewards (they consider only $\theta_v = 1$ for all online v) and vertex weights. They present a $(1 - 1/\mathsf{e})$-competitive algorithm for the special cases of *decomposable probabilities* ($p_{u,v} = p_u p_v$ for every edge u, v) and *vanishing probabilities* ($p_{u,v} \to 0$).

The recent work of Gamlath et al. (2019) studies an offline version of the problem, wherein all of the vertices of the bipartite graph are offline, but the edges are stochastic. For the *probe-commit* model with no limit on the number of probes (i.e., the special case of patience equal to the degree of each vertex), they achieve a competitive ratio of $1 - 1/\mathsf{e}$. Their techniques, like ours, utilize a non-standard LP to upper bound the weight of an optimal matching. However, their LP formulation and solution technique are different from ours.

1.2 Preliminaries and Notation

We use $G = (U, V; E)$ to denote a bipartite graph with vertex set $U \cup V$ and edge set $E \subseteq U \times V$. Let $U = \{u_1, \ldots, u_m\}$ represent offline vertices and $V = \{v_1, \ldots, v_n\}$ represent online vertices. Let w_i denote the *weight* (or value) of offline vertex $u_i \in U$. We assume, wlog, that $w_1 \geq w_2 \geq \ldots w_m$. When discussing the edge weighted case, we let w_{uv} denote the weight of edge (u, v).

For each edge $(u_i, v_j) \in U \times V$, let $p_{i,j}$ denote the given probability that edge (u_i, v_j) exists when probed. When discussing star graphs (with a single online vertex v), we simplify notation and write p_i to denote the probability of edge (u_i, v). We also use $p_{u,v}$ for the given probability of edge (u, v) when indices i and j are not required. For simplicity, we may assume G is the *complete* bipartite graph with $E = U \times V$ by allowing $p_{u,v} = 0$ for nonexistent edges. Thus, when we refer to an edge (u, v) as *incident* to a vertex v, or to u being *adjacent* to or a *neighbor* of v, we mean that $p_{u,v} > 0$ (i.e., that edge (u, v) has a positive probability of existence). We

Adversarial	**Unweighted**	**Vertex-weighted**	**Edge-weighted**
Non-stochastic	0.632 (tight) Karp et al. (1990)	0.632 (tight) Aggarwal et al. (2011)	–
Stochastic rewards	0.5 Mehta and Panigrahi (2012)	? → **0.5**	–
Patience	? → **0.5**	? → **0.5**	–

Random order			
Non-stochastic	0.696 Mahdian and Yan (2011)	0.6534 Huang et al. (2018)	1/e (tight) Kesselheim et al. (2013)
Stochastic rewards	0.632 Borodin et al. (2020)	0.632 Borodin et al. (2020)	1/e (tight) Borodin et al. (2020)
Patience	0.632 Borodin et al. (2020)	? → **0.5**	?

Prophet			
Non-stochastic	0.632 Alaei et al. (2012)	0.632 Alaei et al. (2012)	0.5 Alaei et al. (2012)
Patience	? → **0.632**	? → **0.632**	? → **0.5**

Known IID			
Non-stochastic	0.7299 Brubach et al. (2020)	0.7299 Brubach et al. (2020)	0.705 Brubach et al. (2020)
Stochastic rewards	0.632 Brubach et al. (2020)	0.623 Brubach et al. (2020)	0.623 Brubach et al. (2020)
Patience	0.46 → **0.632** Brubach et al. (2017)	0.46 → **0.632** Brubach et al. (2017)	0.46 → **0.632** Brubach et al. (2017)

Table 1: Landscape of online matching results excluding the unknown patience models we introduce. The bold results with arrows show contributions of this paper. Question marks denote problems where no prior bound was known. If a result follows immediately from the work of a paper, we cite that paper even if the specific result was not mentioned in the paper itself. Our results for unknown patience are not included here since there is no prior work on those models.

are further given a *patience* value θ_j for each online vertex $v_j \in V$ (we sometimes write θ_v for $v \in V$ when subscripts aren't required) that signifies the number of times we are allowed to probe different edges incident on v_j when it arrives. Each edge may be probed at most once and if it exists, we must match it and stop probing (probe-commit model).

We consider the online vertices arriving at positive integer *times*. In the adversarial arrival model, the vertices of $V = \{v_1, v_2, \ldots, v_n\}$ are fixed and the order of their arrival is set by an adversary so as to minimize the expected matching weight. We assume (wlog) that the vertices arrive in the order $v_1, v_2, \ldots, v_n$. When we consider the prophet inequality and known IID arrival settings, V instead specifies a set of vertex *types*, and at time t, a vertex type is randomly chosen (according to a known distribution) from V to arrive; each arrival is independent, so that a given type may arrive multiple times. In these random arrival models, a given time horizon T, denotes the total number of arrivals.

When an online vertex v arrives at time t, we attempt to match it to an available offline vertex. We are allowed to probe edges incident to v_t one-by-one, stopping as soon as an edge (u_i, v_t) is found to exist, at which point the edge is included in the matching and we receive a reward of w_i. We are allowed to probe a maximum of θ_t edges (in the stochastic patience models, θ_k is not known a priori and is discovered only after θ_k failed probes); if θ_k edges are probed and none of the edges exist, then vertex v_k remains unmatched and we receive no reward. If we successfully match v_k to u_i, we say that w_i is the *value* or *reward* of v_k's match; if v_k remains unmatched, we say it has a value or reward of 0.

1.3 Overview

In Section 2, we present algorithms for solving stochastic matching on star graphs—that is, graphs with a single online vertex. We address three different models for the patience of the online vertex: deterministic known patience, stochastic (unknown) patience with a constant hazard rate (i.e., the patience is drawn from a *geometric* distribution), and stochastic patience with an arbitrary distribution (known and given explicitly in the input). To our knowledge, only the deterministic patience has been previously studied. Section 3 presents several LP-based approaches to solving the general online problem, using star graph algorithms (from Section 2) as black boxes.

First, we present a simple online strategy for adversarial arrivals (Section 3.1.1). Then, we present an LP and algorithm for the prophet setting, where each arrival is drawn randomly from a known distribution. In this setting, we present results for both vertex-weighted and edge-weighted variants, as well as an improved competitive ratio for the special case of known IID arrivals (i.e., each arrival is drawn from the same distribution as other arrivals). Finally, we present negative results in Section 4, which show gaps between natural LP relaxations and the performance of an optimal algorithm for the corresponding online problem.

Proofs of all theorems can be found in the full version of the paper.

2 Algorithms for Star Graphs

Here, we present algorithms for offline stochastic matching on star graphs. We can view this equivalently as the online stochastic matching problem in the special case where there is a single online arrival (i.e., $|V| = 1$). In all cases, we consider *weighted* graphs. Note that the edge-weighted and vertex-weighted problems are equivalent on star graphs.

2.1 Known Patience

An algorithm to solve the known patience case via dynamic program was given in Purohit et al. (2019) (see section 2.1 of that paper, i.e. the $k = 1$ case of the hiring problem they study). We present this briefly here but do not discuss it in detail.

Recall that we assume wlog that $w_1 \geq w_2 \geq \cdots \geq w_m$. To simplify notation, we write p_i for $p_{u_i,v}$. We then define $f(i,\theta)$ to be the maximum possible expected value if we may only match v to one of the neighbors in $\{v_i, \ldots, v_m\}$ (i.e., neighbors i through m).

$$f(i,\theta) = \max\{p_i w_i + (1-p_i) f(i+1, \theta-1), \\ f(i+1,\theta)\} \tag{1}$$

with $f(i, 0) = 0$.

This dynamic program immediately implies the following fact about the known patience setting:

Theorem 2.1. *There exists an algorithm for the stochastic matching with known patience problem on vertex-weighted star graphs that finds the optimal probing strategy in polynomial time.*

2.2 Constant Hazard Rate

As before, suppose we have the star graph $S = (U, V, E)$ with $U = \{u_1, u_2, \ldots, u_m\}$ and $V = \{v\}$. In this setting, the patience is random and unknown, with a constant "hazard rate." For v, we are given the "survival probability," denoted by r. When an attempted match is unsuccessful, the central vertex v, with probability r, remains available for another match attempt. The "hazard rate" $1-r$ is the probability that v becomes unavailable after a failed probe. Equivalently, the patience is drawn from a negative binomial distribution, $\theta_v \sim \mathrm{NB}(1, 1-r)$.

We briefly describe the intuition behind our approach: Imagine we could repeatedly probe an edge—with the success of each probe being independent of other probes—until it matches or the patience of the center vertex is exhausted. Let v_e be the expected weight achieved by repeatedly probing an edge in this way. So, $v_e = w_e p_e \sum_{i=0}^{\infty} r^i (1-p_e)^i$, which equals $\frac{w_e p_e}{1-r(1-p_e)}$ since $r(1-p_e) < 1$.

When $r = 0$, we only get one probe and $v_e = w_e p_e$. When $r = 1$, there is no patience constraint and $v_e = w_e$. In both extreme cases, probing in decreasing order of v_e is the optimal strategy. It turns out that probing in decreasing order of v_e is also optimal for any fixed r. This is stated formally in Theorem 2.2.

Theorem 2.2. *Probing in decreasing order of* $v_e = \frac{w_e p_e}{1-r+rp_e}$ *is optimal for all* $r \in [0,1]$ *and* $p_e \in (0,1]$.

2.3 Arbitrary Patience Distributions

We now address the case where the patience is stochastic (and unknown) and can follow an arbitrary distribution. We use an LP-based approach for this problem. We denote by q_θ the probability that the patience of the online vertex v is *at least* θ. Notice that $1 = q_1 \geq q_2 \geq \ldots \geq q_n$. Our approach utilizes the LP (2) described in the next paragraph. The variables are $x_{j\theta}$, corresponding to the probability of attempting to match with j on the θ^{th} attempt. The value s_θ represents the probability that the online vertex is available for a θ^{th} match attempt, meaning its patience is at least θ and all previous match attempts were unsuccessful. This can be calculated from the $x_{j\theta}$, q_θ, and p_j values. Our Linear Program (2) has objective function

$$\max \sum_{j=1}^{n} w_j p_j \sum_{\theta=1}^{n} x_{j\theta} \tag{2}$$

subject to the following constraints: (2a) $\sum_{\theta=\theta'}^{n} x_{j\theta} \leq s_{\theta'}$ for all $j \in \{1, 2, \ldots, n\}$ and all $\theta' \in \{1, 2, \ldots, n\}$ (2b) $\sum_{j=1}^{n} x_{j\theta} \leq s_\theta$ for all $\theta \in \{1, 2, \ldots, n\}$ (2c) $x_{j\theta} \geq 0$ for all $j \in \{1, 2, \ldots, n\}$ and $\theta \in \{1, 2, \ldots, n\}$ (2d) $s_1 = 1$ (2e) $s_\theta = \frac{q_\theta}{q_{\theta-1}} \left(s_{\theta-1} - \sum_{j=1}^{n} p_j x_{j,\theta-1} \right)$ for all $\theta \in \{1, 2, \ldots, n\}$.

Constraint (2a) says that each offline vertex j can be probed at most once across attempts numbered $\theta', \theta'+1, \ldots, n$ when conditioned on still being available at attempt θ'. Note that this family of constraints is novel in comparison to similar time-indexed LP's which have appeared in the literature (Ma, 2014), in that there is a constraint for the sum starting at every attempt θ', instead of just a single constraint where $\theta' = 1$. Constraint (2b) says that we may only attempt the θ^{th} match if the online vertex is still available for a θ^{th} attempt. Constraints (2d)–(2e) ensure that the values of s_θ are updated correctly, where the fraction $\frac{q_\theta}{q_{\theta-1}}$ is understood to be 0 if both q_θ and $q_{\theta-1}$ are 0. We can see that this LP upper bounds the optimal algorithm, since taking $x_{j\theta}$ to be the probability of the algorithm probing j on attempt θ for all j and θ, we get a feasible solution to the LP with objective value equal to the algorithm's expected weight matched.

Our algorithm is simple: we solve LP (2) to get an optimal solution x^*, along with the values s^*. Then, when making the θ^{th} probe, choose each offline vertex $j = 1, \ldots, n$ with probability $x^*_{j\theta}/s^*_\theta$ (note that if $s^*_\theta = 0$ then $x^*_{j\theta} = 0$), which defines a proper probability distribution by (2b). If a vertex j is chosen to be probed, but has already been unsuccessfully probed in a previous attempt, we "simulate" probing j instead, and *terminate* with no reward if the simulated probe is successful. This simulation technique, motivated by Brubach et al. (2017), is important in ensuring that the probability of surviving to the θ'th attempt is consistent with the LP value s^*_θ.

Theorem 2.3. *The online algorithm based on LP (2) is a 1/2-approximation for the star graph probing problem, for an arbitrary patience distribution which is given explicitly.*

The analysis in Theorem 2.3 is tight. We further note that the result of Theorem 2.3 compares to a benchmark (LP (2)) that does *not* know the full realization of the patience values in advance. This is necessary, since Theorem 4.3 states that comparing to a benchmark which knows the patience in advance leads to arbitrarily bad competitive ratios.

3 Algorithms for Online Matching

3.1 Vertex-Weighted, Adversarial Arrivals

We present a greedy algorithm which achieves a 0.5-approximation for online matching with vertex weights, stochastic rewards, and patience constraints in the adversarial arrival model. In this setting, the vertices of V arrive in an online fashion. If vertex v is the t^{th} vertex to arrive online, we say v *arrives at time t*.

Algorithm 1: Use Star Graph Black Box 1 to greedily match arriving vertices

```
Function AdvGreedy(U, V, p, w):
    for Arriving vertex v ∈ V do
        (i*_1, ..., i*_m) ← StarBB(v, p, w)
        for θ := 1 to m do
            Probe edge (u_{i*_θ}, v)
```

Our algorithm works by simply using a black box procedure for the star graph induced by the online arrival v and all of its still unmatched neighbors $U' \subseteq U$. Let $\mathsf{ALG}(G)$ denote the expected size of the matching produced by this algorithm on the graph G. Let $\mathsf{OPT}(G)$ denote the expected size of the matching produced by an optimal *offline* algorithm. Our main result here is:

Theorem 3.1. *Given a κ-approximate black box for solving star graphs, Algorithm 1 achieves a competitive ratio of 0.5κ; that is, for any bipartite graph G, $\frac{\mathsf{ALG}(G)}{\mathsf{OPT}(G)} \geq 0.5\kappa$.*

To prove this, we first present an LP which provides an upper bound on the offline optimal.

3.1.1 An LP upper bound on OPT

The typical LP relaxation for our problem has objective function

$$\max \sum_{u \in U} \sum_{v \in V} x_{u,v} p_{u,v} w_u \tag{3}$$

for $0 \leq x_{u,v} \leq 1$, subject to the following constraints: (3a) $\sum_{v \in V} x_{u,v} p_{u,v} \leq 1$ for all $u \in U$ (3b) $\sum_{u \in U} x_{u,v} p_{u,v} \leq 1$ for all $v \in V$ (3c) $\sum_{u \in U} x_{u,v} \leq \theta_v$ for all $v \in V$. For our analysis, we will also add an additional constraint to tighten the upper bound provided by the LP. We slightly abuse notation and write $\mathsf{OPT}(U', v)$ to denote the optimal strategy for a star graph $(U', \{v\}, U' \times \{v\})$ (recall that we can compute $\mathsf{OPT}(U', v)$ optimally with dynamic program (1). To strengthen the LP, we add the constraint (3d) $\sum_{u \in U'} x_{u,v} p_{u,v} w_u \leq \mathsf{OPT}(U', v)$ for all $U' \subseteq U$ and $v \in V$

We first observe that this LP gives an upper bound on the optimal solution. The proof of this is not difficult, and is omitted here.

Lemma 3.2. *For any bipartite graph G, $\mathsf{OPT}_{\mathsf{LP}}(G) \geq \mathsf{OPT}(G)$.*

Then, we use this LP in order to analyze our algorithm. We establish the following lemma, stating that our algorithm achieves an expected matching that is at least half the LP value.

Lemma 3.3. *For any bipartite graph G, $\mathsf{ALG}(G) \geq 0.5\mathsf{OPT}_{\mathsf{LP}}(G)$.*

The proof utilizes a charging argument; it is somewhat long and is omitted here. Using this, we are able to establish our main result for this setting:

Proof of Theorem 3.1. By Lemmas 3.2 and 3.3, we have

$$\mathsf{ALG}(G) \geq 0.5\kappa \mathsf{OPT}_{\mathsf{LP}}(G) \geq 0.5\kappa \mathsf{OPT}(G)$$

□

This analysis is for the known patience problem. The same argument can be used to show that for unknown patience, a κ-approximate solution to the star graph problem yields a $\kappa/2$-competitive matching in the online setting with adversarial arrivals. Thus, we can achieve a 1/2 competitive ratio for constant hazard rates and 1/4 for arbitrary patience distributions.

3.2 Edge-Weighted in the Prophet Setting

With edge weights, no constant competitive ratio is possible under adversarial arrivals. Thus, we consider a different arrival model, which we refer to as the *prophet arrival model*. In this model, V specifies a set of possible arrival *types*; each arrival takes on one of these types randomly, according to a known distribution. The probabilities at each arrival are independent of previous arrivals, and the distribution over possible types can be different at each arrival.

For $t = 1, 2, \ldots, T$ and $v \in V$, denote by q_{tv} the probability that the vertex arriving at time t will be of type v. For convenience, we denote by $q_v = \sum_{t=1}^{T} q_{tv}$ the expected number of arrivals of a vertex of type v.

We employ a new exponential-sized LP relaxation. In this LP, the variables correspond to *policies* for probing an arriving online vertex. A deterministic policy π for matching any online vertex type v is characterized by a permutation of some *subset of* U. The policy specifies the strategy of attempting to match v to vertices of U in the order given by π, until either a probe is successful, all vertices in π are attempted, or the patience of v is exhausted. Let $\mathcal{P}$ denote the set of all deterministic policies.

We present our LP in (4) below. We let $p_{uv}(\pi)$ denote the probability that an online arrival of type v is matched to offline vertex u when following policy π, assuming that all vertices of π are still unmatched. The decision variables in the LP are given by $x_v(\pi)$, and we let $\mathsf{OPT}_{\mathsf{LPP}}$ denote the true optimal objective value of the exponential-sized LP.

$$\mathsf{OPT}_{\mathsf{LPP}} = \max \sum_{v \in V} \sum_{\pi \in \mathcal{P}} x_v(\pi) \sum_{u \in U} p_{uv}(\pi) w_{uv} \tag{4}$$

subject to the following constraints: (4a) $\sum_{v \in V} \sum_{\pi \in \mathcal{P}} x_v(\pi) p_{uv}(\pi) \leq 1$ for all $u \in U$ and (4b) $\sum_{\pi \in \mathcal{P}} x_v(\pi) = q_v$ for all $v \in V$, and (4c) $x_v(\pi) \geq 0$ for all $v \in V$ and $\pi \in \mathcal{P}$.

We can interpret $x_v(\pi)$ as the expected number of times policy π will be applied to an online vertex of type v. Constraint (4a) says that in expectation each offline vertex u can be matched at most once. Constraint (4b) comes from the fact that exactly one policy (possibly the policy which makes zero probes) must be applied on each arriving vertex of type v. It follows from standard techniques (see e.g. Lemma 9 in Bansal et al. (2010)) that even for an offline clairvoyant, who knows the realized types of arrivals in advance, if we let $x_v(\pi)$ denote the expected number of times it applies policy π on an online vertex of type v, then this forms a feasible solution to the LP with objective value equal to the clairvoyant's expected weight matched. Therefore, if we can bound the algorithm's expected weight matched relative to $\mathsf{OPT}_{\mathsf{LPP}}$, then this would yield a competitive ratio guarantee.

The LP (4), although exponential-sized, can be solved in polynomial time by the Ellipsoid Method, as shown in the full version of the paper.

3.2.1 Algorithm and Analysis based on Exponential-sized LP

We now show how to use the LP (4) to design a $\kappa/2$-competitive online algorithm, given a feasible LP solution $x_v^*(\pi)$ which is at least $\kappa \cdot \mathsf{OPT}_{\mathsf{LPP}}$, for some $\kappa \leq 1$. For each vertex $u \in U$, let $w_u^* = \sum_{v \in V} w_{uv} \sum_{\pi \in \mathcal{P}} p_{uv}(\pi) x_v^*(\pi)$ denote the expected reward of matching u according to the assignment x^*. Notice that the objective value of the given solution is $\sum_{u \in U} w_u^*$, which is at least $\kappa \cdot \mathsf{OPT}_{\mathsf{LPP}}$. Using this notation, our algorithm is given in Algorithm 2. By LP constraint (4b), we have $\sum_{\pi \in \mathcal{P}} x_v^*(\pi)/q_v = 1$, so the probability distribution over policies defined in Algorithm 2 is proper. We also note that since at most polynomially many variables $x_v^*(\pi)$ will be nonzero, this distribution has polynomial-sized support and can be sampled from in polynomial time.

Algorithm 2: Random Arrivals from Known Distributions (Prophet Arrivals)

```
Function OnlineMatch(U, V, p, w):
    for t = 1 to T do
        Online vertex v_t, of type v ∈ V, arrives
        Choose a policy π = (π_1, π_2, ..., π_ℓ) with
          probability x*_v(π)/q_v
        for i := 1 to ℓ do
            if w_{π_i,v} < w*_{π_i}/2 then
                Skip to next i
            else if π_i is unmatched then
                Probe edge (π_i, v_t) and match if
                  successful for reward w_{π_i,v_t}
            else
                Simulate probing (π_i, v_t). If
                  successful, move to next arrival
                  without matching v_t.
```

Theorem 3.4. *Under general edge weights and known arrival distributions, Algorithm 2 is $\kappa/2$-competitive for the online bipartite matching with patience problem, assuming we are given a solution to LP (4) with objective value at least $\kappa \cdot \mathsf{OPT}_{\mathsf{LPP}}$.*

When we can solve the LP (4) exactly, $\kappa = 1$. An immediate consequence of Theorem 3.4, in conjunction with our results from Sections 2.1 and 2.2, is that if all patiences are either deterministic or stochastic with constant hazard rate, then we have a 1/2-competitive polynomial-time algorithm for online bipartite matching with patience. For general stochastic patiences, $\kappa = 1/2$ and Theorem 3.4 imply an 1/4-competitive algorithm.

3.3 Edge-Weighted with IID Arrivals

In the case of IID arrivals, i.e., where $q_{1v} = q_{2v} = \cdots = q_{Tv}$ for all vertex types $v \in V$, a slight modification to Algorithm 2 yields a competitive ratio of $(1 - 1/e)\kappa$ when given a feasible solution to LP 4 that is at least $\kappa \cdot \mathsf{OPT}_{\mathsf{LPP}}$ of optimal.

The only change to the algorithm from the previous non-IID setting is that for IID arrivals, we do not skip probing vertices $u \in U$ when $w_{uv} < w_u^*/2$.

Theorem 3.5. *Under general edge weights and known IID arrivals, Algorithm 3.3 is $\kappa(1 - 1/\mathsf{e})$-competitive for the online bipartite matching with patience problem, assuming we are given a solution to LP (4) with objective value at least $\kappa \cdot \mathsf{OPT}_{\mathsf{LPP}}$.*

Theorem 3.5 implies a $(1 - 1/\mathsf{e})$-competitive algorithm when all patiences are deterministic or stochastic with constant hazard rate, and a $\frac{1}{2}(1 - 1/\mathsf{e})$-competitive algorithm for general stochastic patiences.

3.4 Vertex-Weighted, Prophet Setting

With a new analysis, we can show that the algorithm from section 3.3 still achieves a competitive ratio of $1 - 1/\mathsf{e}$ in the prophet (non-identical) setting in the case of *vertex weights*.

Theorem 3.6. *Under vertex-weights and known arrival distributions, Algorithm 3.3 is $\kappa(1 - 1/e)$-competitive for the online bipartite matching with patience problem, assuming we are given a solution to LP (4) with objective value at least $\kappa \cdot \mathsf{OPT}_{\mathsf{LPP}}$.*

4 Negative Results

4.1 Stochasticity Gap

The *stochasticity gap* is a fundamental gap in Linear Programming relaxations for stochastic problems which replace probabilities with deterministic fractional weights. The notion was first discussed informally in Brubach et al. (2017), and was later also observed by Purohit et al. (2019) (where they referred to it as a "probing gap"). When these LP relaxations are used as upper bounds on the offline optimal solution, or as a benchmark for the competitive ratio, the stochasticity gap represents a barrier to the best achievable competitive ratio. One interpretation of such a result is that better competitive ratios are not possible. However, one may alternatively view it as a result showing the limitations of using a particular LP as a benchmark for competitive ratios.

We present a stochasticity gap for a common LP relaxation of the online bipartite matching problem with stochastic rewards. Recall from Section 3.1.1 that LP (3) with the constraints (3a)–(3c) (and excluding our additional family of constraints (3d)) is a standard LP relaxation for bipartite matching with (known) patience constraints and adversarial arrivals. This is essentially an extension of the "Budgeted Allocation" LP from Mehta and Panigrahi (2012) to include the patience constraints.

A fairly simple example and analysis can establish a stochasticity gap of $1 - 1/\mathsf{e}$ for this LP. A more complicated analysis establishes our main result for the stochasticity gap of this LP:

Theorem 4.1. *The LP given by the objective function* (3) *and constraints* (3a)–(3c) *has a stochasticity gap of* $\lessapprox 0.544$.

4.2 0.5 Upper Bound for SimpleGreedy

As defined in Mehta and Panigrahi (2012), an *opportunistic* algorithm for the *Stochastic Rewards* setting is one which always attempts to probe an edge incident to an online arriving vertex $v \in V$ if one exists. The work of Mehta and Panigrahi (2012) showed that in the unweighted *Stochastic Rewards* ($\theta_v = 1$ for all online vertices $v \in V$) problem, any opportunistic algorithm achieves a competitive ratio of 1/2. The simplest opportunistic algorithm is the one which, when $v \in V$ arrives online, chooses a neighbor $u \in U$ of v arbitrarily and probes the edge (u, v). We call this algorithm "SimpleGreedy". Since SimpleGreedy is opportunistic, the result of Mehta and Panigrahi (2012) shows that SimpleGreedy achieves a competitive ratio of at least 1/2; Theorem 4.2 shows that this is tight even when restricted to small, uniform p.

Theorem 4.2. *There exists a family of unweighted graphs under stochastic rewards and adversarial arrivals for which* SimpleGreedy *achieves a competitive ratio of at most* 1/2 *even when all edges have uniform probability* $p = O(1/n)$.

4.3 Hardness of Unknown Patience

We now show that matching on a star graph (i.e. a single online arrival) with a random unknown patience, in general, should not be compared to a benchmark that knows the realization of the patience in advance. This is because if we use such a benchmark, the competitive ratio may be arbitrarily bad. We formalize this statement in Theorem 4.3.

Theorem 4.3. *When compared to a benchmark that knows the patience in advance, the competitive ratio is* $o(1)$.

A consequence is that a naive LP formulation of the unknown patience problem, which simply replaces the patience with its expected value, cannot be used to define the competitive ratio in a meaningful way (or, even, to upper bound it). Further, the problem of unknown patience where the patience value is not determined randomly but is instead fixed in advance by an adversary, cannot have an $O(1)$ competitive ratio when compared to a benchmark that knows the patience in advance.

Acknowledgments

Brian Brubach was supported in part by NSF awards CCF-1422569 and CCF-1749864, and by research awards from Adobe.

Nathaniel Grammel was supported in part by NSF award CCF-1918749, and by research awards from Amazon and Google.

Aravind Srinivasan was supported in part by NSF awards CCF-1422569, CCF-1749864, and CCF-1918749, as well as research awards from Adobe, Amazon, and Google.

References

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*, pages 1–12, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. ISBN 978-3-662-48350-3.

Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the twenty-second annual*

ACM-SIAM symposium on Discrete Algorithms, pages 1253–1264. SIAM, 2011.

Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online prophet-inequality matching with applications to ad allocation. *Proceedings of the ACM Conference on Electronic Commerce*, 06 2012. doi: 10.1145/2229012.2229018.

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. In *Algorithms – ESA 2010: 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part II*, pages 218–229, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15781-3.

Allan Borodin, Calum MacRury, and Akash Rakheja. Bipartite stochastic matching: Online, random order, and i.i.d. models, 2020.

Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. New algorithms, better bounds, and a novel model for online stochastic matching. *European Symposium on Algorithms (ESA)*, 2016.

Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Attenuate locally, win globally: An attenuation-based framework for online stochastic matching with timeouts. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, 2017.

Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Online stochastic matching: New algorithms and bounds. *Algorithmica*, 2020. URL http://arxiv.org/abs/1606.06395.

Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. Edge-weighted online bipartite matching. In *Foundations of Computer Science (FOCS)*. IEEE, 2020.

Jon Feldman, Nitish Korula, Vahab Mirrokni, S Muthukrishnan, and Martin Pál. Online ad assignment with free disposal. In *Internet and network economics*, pages 374–385. Springer, 2009.

Buddhima Gamlath, Sagar Kale, and Ola Svensson. Beating greedy for stochastic bipartite matching. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 2841–2854, 2019. doi: 10.1137/1.9781611975482.176.

Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 982–991, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

Vineet Goyal and Rajan Udwani. Online Matching with Stochastic Rewards: Optimal Competitive Ratio via Path Based Formulation. In *Proc. ACM Conference on Economics and Computation*, 2020.

Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals. *CoRR*, abs/1804.07458, 2018. URL http://arxiv.org/abs/1804.07458.

Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *European Symposium on Algorithms (ESA)*, pages 589–600. Springer, 2013.

Will Ma. Improvements and generalizations of stochastic knapsack and multi-armed bandit approximation algorithms. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1154–1163. Society for Industrial and Applied Mathematics, 2014.

Will Ma and David Simchi-Levi. Algorithms for online matching, assortment, and pricing with tight weight-dependent competitive ratios. *Operations Research*, 2020.

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*, pages 597–606. ACM, 2011.

Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2012. ISSN 1551-305X.

Aranyak Mehta and Debmalya Panigrahi. Online matching with stochastic rewards. In *Foundations of Computer Science (FOCS)*, pages 728–737. IEEE, 2012.

Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2015.

Manish Purohit, Sreenivas Gollapudi, and Manish Raghavan. Hiring under uncertainty. In *International Conference on Machine Learning*, pages 5181–5189. PMLR, 2019.