MATHEMATICS OF OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0364-765X | EISSN 1526-5471 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Prophet Matching with General Arrivals *

Tomer Ezra
Tel Aviv University, tomer.ezra@gmail.com

Michal Feldman
Tel Aviv University and Microsoft Research, michal.feldman@cs.tau.ac.il

Nick Gravin
ITCS, Shanghai University of Finance and Economics, nikolai@mail.shufe.edu.cn

Zhihao Gavin Tang
ITCS, Shanghai University of Finance and Economics, tang.zhihao@mail.shufe.edu.cn

We provide prophet inequality algorithms for online weighted matching in general (non-bipartite) graphs, under two well-studied arrival models, namely edge arrival and vertex arrival. The weights of the edges are drawn from a priori known probability distribution. Under edge arrival, the weight of each edge is revealed upon arrival, and the algorithm decides whether to include it in the matching or not. Under vertex arrival, the weights of all edges from the newly arriving vertex to all previously arrived vertices are revealed, and the algorithm decides which of these edges, if any, to include in the matching. To study these settings, we introduce a novel unified framework of batched-prophet inequalities that captures online settings where elements arrive in batches. Our algorithms rely on the construction of suitable online contention resolution scheme (OCRS). We first extend the framework of OCRS to batched-OCRS, we then establish a reduction from batched-prophet inequality to batched-OCRS, and finally we construct batched-OCRSs with selectable ratios of 0.337 and 0.5 for edge and vertex arrival models, respectively. Both results improve the state of the art for the corresponding settings. For vertex arrival, our result is tight. Interestingly, a pricing-based prophet inequality with comparable competitive ratios is unknown.

Key words: prophet inequality; online matching; online stochastic matching; online contention resolution scheme
MSC2000 subject classification: Primary: 68W27; secondary: 90C27, 68Q87

---

1. Introduction Online matching is a central problem in the area of online algorithms, and is extensively used in economics to model rapidly appearing online markets. Some prominent applications include matching platforms for ride sharing, healthcare (e.g., kidney exchange programs), job search, dating, and internet advertising [44, 15, 8, 43]. As many of the online platforms accumulate

* A preliminary version of this paper appeared in the ACM Conference on Economics and Computation 2020 under the title "Online Stochastic Max-Weight Matching: prophet inequality for vertex and edge arrival models". This work is supported by Science and Technology Innovation 2030 -"New Generation of Artificial Intelligence" Major Project No.(2018AAA0100903), Innovation Program of Shanghai Municipal Education Commission, Program for Innovative Research Team of Shanghai University of Finance and Economics (IRTSHUFE) and the Fundamental Research Funds for the Central Universities. The first two authors are partially supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 866132), and by the Israel Science Foundation (grant number 317/17). The last author is partially supported by National Natural Science Foundation of China (NSFC) 61902233.

huge amounts of statistical data and use it heavily in their online decision making, it is natural to study online matching in Bayesian settings; i.e., assuming a priori knowledge of the probability distribution from which values are drawn.

The study of optimal (or nearly optimal) online decisions in Bayesian settings has been extensively studied within the *prophet inequality* paradigm. Prophet inequality appeared first as a fundamental result in optimal stopping theory [38, 39, 47], and was later extended to online selection problems under more complex feasibility constraints, such as uniform matroids [29], general matroids [37], intersection of matroids [37], general downward-closed feasibility constraints [45], and most relevant to our present work: matching in graphs [3, 21, 26].

In a *prophet matching* problem in bipartite graphs with one-sided vertex arrival, edge weights are drawn from known probability distributions, and upon the arrival of a vertex v, the weights of all edges from v to its neighbors are revealed. An immediate and irrevocable decision should be made by the online algorithm regarding whether to match v and to which vertex.

The performance of an online algorithm is measured by its *competitive ratio*, namely the ratio between the expected total weight of the selected matching and the expected total weight of the optimal matching.

For one-sided vertex arrival in bipartite graphs, Feldman et al. [21] devised a 1/2-competitive algorithm, for an adversarial arrival order, using a pricing-based approach.[1] In the edge arrival model, edges arrive in an adversarial order, and edge weights are revealed upon arrival. Clearly, the edge arrival model is harder than the vertex arrival model, since edge weights are revealed one by one, whereas in the vertex arrival model, some edge weights are revealed simultaneously. Indeed, Gravin and Wang [27] showed that no prophet algorithm for this setting achieves a better competitive ratio than 4/9, even in bipartite graphs (this upper bound was later improved to 3/7 by Alon et al. [4]). On the positive side, Gravin and Wang [27] devised a pricing/threshold-based algorithm that gives a competitive ratio of 1/3 for matching in bipartite graphs.

The aforementioned studies focus on matching in bipartite graphs, and under vertex arrival they further focus on one-sided arrival. In reality, many matching applications cannot be captured by these models. For example, matching in ride-sharing platforms is better captured by two-sided bipartite graphs, where both drivers and passengers arrive dynamically. This is also the case for buyers and sellers in e-commerce applications, ad slots and advertisers in ad auctions, and jobs and workers in online labor markets. Moreover, other applications cannot be captured by bipartite matching at all, and are better captured by matching in general graphs. This is the case, for example, in exchange platforms such as kidney exchange, where every vertex corresponds to a patient-donor pair, and all vertices play the same role, with no distinction to "left" and "right" sets.

A natural problem arises: *Do the competitive ratios obtained for prophet matching in bipartite graphs extend to general graphs?* This is the problem we study in this work.

Interestingly, a similar evolution took place with respect to the seminal paper of Karp et al. [33]. The original paper introduced the problem of online matching in the fully adversarial model, and studies it under one-sided vertex arrival in bipartite graphs. Follow-up work by Gamlath et al. [25], Wang and Wong [48] consider this problem under more general arrival models, including edge arrival in bipartite and general graphs, and vertex arrival under both two-sided arrival in bipartite graphs and general graphs. By analogy, our work introduces the same extensions with respect to matching in prophet inequality setting (e.g., Bayesian).

1.1. Our Contribution Table 1 presents our results in the context of previous results. The top row corresponds to unweighted graphs under the fully adversarial model (i.e., no Bayesian information). The bottom row corresponds to weighted graphs under the Bayesian model (i.e., our setting). The left and right columns correspond to the edge- and vertex-arrival models, respectively.

[1] For the case of known or random arrival order, a 1/2-competitive algorithm has been devised by [3].

	Edge Arrival	Vertex Arrival	
	Bipartite / Any graphs	One Sided (Bipartite graphs)	Two Sided (Bipartite) / Any graphs
Worst-case (unweighted)	$\frac{1}{2}$ (tight) bipartite & any graphs [25]	$1-\frac{1}{e}$ (tight) [33]	$\geqslant \frac{1}{2}+\Omega(1)$(any graph) [25] $\leqslant 0.591$(bipartite graphs) [9]
Bayesian (weighted)	$\geqslant \frac{1}{3}$ (bipartite graphs) [27] $\leqslant \frac{4}{9}$ (bipartite graphs) [27] $\geqslant$ **0.337 (any graph)** (Theorem 4)	$\frac{1}{2}$ (tight) [21]	$\geqslant \frac{1}{2}$ **(any graph)** (Theorem 2) $\leqslant \frac{1}{2}$ (tight) one sided arrival bipartite graphs

Table 1. Competitive ratios for online matching: previous and new results. New results are indicated in bold.

The right column is further divided to one-sided arrival in bipartite graphs and more general models (i.e., 2-sided arrival in bipartite graphs, and general arrival in general graphs).

Matching with vertex arrival. We devise a $\frac{1}{2}$-competitive prophet inequality for general (not necessarily bipartite) graphs. This result is tight (a matching upper bound is derived from the classical prophet inequality problem). Moreover, our competitive ratio holds also with respect to the stronger benchmark of the optimal fractional matching. Unlike bipartite graphs, the optimal fractional matching in general graphs may have a strictly higher weight than any integral matching.

An interesting implication of this result is that in the Bayesian setting there is no gap between the competitive ratio that can be obtained under the 1-sided and 2-sided vertex arrival models in bipartite graphs, or even under vertex arrival in general graphs. This is in contrast to the non-Bayesian (worst case) online model, where there is a gap between 1-sided and 2-sided vertex arrivals (see Table 1).

Matching with edge arrival. We construct a 0.337-competitive prophet inequality for general (not necessarily bipartite) graphs, under the edge arrival model. This improves upon the 1/3-competitive prophet inequality constructed by Gravin and Wang [27] and Kleinberg and Weinberg [37]. While these previous studies take a pricing/threshold-based approach, we use a different approach, namely OCRS (see details below). We show that the $\frac{1}{3}$-competitive ratio for bipartite graphs can be obtained by a simple OCRS construction. Moreover, our OCRS construction generalizes to general graphs[2]. We further improve the competitive ratio to 0.337 by constructing a better OCRS, which requires more subtle analysis. These results hold against the even stronger benchmark of the *ex-ante* optimal solution that satisfies fractional matching constraints (similar to the observation in Lee and Singla [41]). Additionally, these results apply to multigraphs as well, see Appendix G.

1.2. Our Techniques To study prophet inequality for matching, we introduce a unified framework of *batched-prophet inequalities* that captures online settings where elements arrive in *batches* rather than one by one. For example, in the vertex arrival model, upon the arrival of a vertex v, the weights of all edges from v to previous vertices are revealed simultaneously. Unlike the classical setting, where the decision in every step is binary (accept/reject), in batched settings, a complex online decision should be made in each step, based on the corresponding feasibility constraints. For example, in the matching example, the online algorithm should decide whether to match the arriving vertex v to a previous vertex, and if so, to which one.

[2] Note, however, that unlike Gravin and Wang [27], the OCRS-based algorithm is adaptive.

Unlike [37, 27, 21], who take a "pricing/charging" approach, our solution takes a different approach, related to a technique known as Online Contention Resolution Schemes (OCRS) [11, 23], which we extend to batched settings. Our techniques proceed in three steps: (i) we extend the framework of OCRS to *batched-OCRS*. (ii) we establish a *reduction* from batched-prophet inequality to batched-OCRS (where selectable ratios of OCRS translate to prophet inequalities with identical competitive ratios). (iii) we construct batched-OCRSs with good selectable ratios. In particular, we construct batched-OCRS with selectable ratios 0.337 and 0.5 for edge- and vertex-arrival models, respectively.

Contention Resolution Schemes (CRS) were introduced by Chekuri et al. [11] as a powerful rounding technique in the context of submodular maximization. The CRS framework was extended to the OCRS framework for online stochastic selection problems by Feldman et al. [23], who provided OCRSs for different problems, including intersections of matroids and matchings, and showed applicability to prophet inequality [23, 41]. Specifically, for the matching feasibility constraint, Feldman et al. [23] constructed a $\frac{1}{2e}$-OCRS that implies a $\frac{1}{2e}$-competitive algorithm for prophet matching under edge arrival.

We devise a general reduction from batched prophet inequality to batched-OCRS for any downward-closed feasibility constraint. This general reduction implies that to get prophet inequalities with a certain competitive ratio, it suffices to construct an OCRS with the same selectable ratio. Given this reduction, it suffices to construct batched OCRS for our matching problems. We do so for both the edge- and vertex-arrival models. (While the batched setting captures both arrival models, for the edge arrival model the standard OCRS suffices.)

The OCRS approach is not as common as the pricing approach in prophet inequality settings. We note that the earlier algorithms of Chawla et al. [10] and Alaei [2], when applied to the classic prophet inequality setting, become a simple $\frac{1}{2}$-competitive algorithm that is indeed a $\frac{1}{2}$-OCRS. These algorithms also appear to be closer in spirit to our OCRS approach than to the more recent papers on prophet inequality (e.g., [37, 21, 27, 16, 42, 18]). Specifically, when restricted to the setting of selecting one item, our algorithm is the same as Alaei's algorithm [2].

One of the reasons that OCRSs are not as prevalent in prophet inequality settings under more general feasibility constraints is that the formal definition of OCRS is not specifically tailored for prophet inequalities. As a result, the approximation factors that are obtained by the OCRS approach are not as tight. For example, the original OCRS introduced by Feldman et al. [23] for matching feasibility constraint achieves a competitive ratio of $\frac{1}{2e}$, whereas even a non-adaptive pricing-based algorithm achieves the much better ratio of $\frac{1}{3}$ [27].

Indeed, these OCRSs are usually designed to work against a strong almighty adversary, who controls the arrival order of the elements and knows in advance the realization of the instance and the random bits of the algorithm. The OCRSs we construct in this work are better tailored to the prophet inequality setting as they are designed against a weaker oblivious adversary, who can select an arbitrary arrival order, but does not observe the algorithm's decisions and the realization of the instance.

Our $\frac{1}{3}$- and $\frac{1}{2}$-selectable OCRSs for the edge- and vertex arrival models, respectively, are surprisingly simple and intuitive. And yet, the latter OCRS already gives a tight competitive ratio for the vertex arrival model. Even more surprisingly, at the time of writing this paper, no pricing-based approach is known to match the $\frac{1}{2}$-competitive guarantee attainable by the OCRS for the vertex-arrival model. Moreover, several natural attempts of generalizing the pricing scheme in Feldman et al. [21] fail miserably, even for bipartite graphs. For example, one natural generalization would be to set the price on a new vertex v to be half of the expected contribution of the future edges incident to v to the optimum matching. As it turns out, this pricing scheme achieves a competitive ratio as small as $\frac{1}{4}$. This is demonstrated in Appendix F. We note, however, that formulating a general OCRS framework for batched arrivals of elements is not as trivial as it might seem at a first glance. We give an example in Appendix E illustrating why a simpler and apparently more natural than ours extension of OCRS to the setting with batched arrivals can be problematic.

On the other hand, the $\frac{1}{3}$-OCRS for the edge arrival model is based on a simple union bound which still leaves some room for improvement. We improve the ratio of $\frac{1}{3}$ to 0.337 by bounding the negative correlation for any pair of events that vertex u and vertex v are matched at the arrival time of edge (uv). This is the most technical part of the paper.

1.3. Related Work

Prophet inequality. Prophet inequality is highly relevant to the algorithmic mechanism design literature. Hajiaghayi et al. [29] were the first to realize the applicability of the prophet inequality framework within mechanism design applications. Later, Chawla et al. [10] applied the prophet inequality framework to the design of sequential posted price mechanisms that give approximately optimal revenue in a Bayesian multi-parameter unit-demand setting (BMUD). An important ingredient in their result is the first constant competitive (specifically, $\frac{4}{27}$) prophet inequality for the online weighted matching problem with edge arrival in a bipartite graph. Kleinberg and Weinberg [37] introduced a general combinatorial prophet inequality for a broad class of Bayesian selection problems, where the feasible set is represented as an intersection of k matroids. They found a $\frac{1}{4k-2}$-competitive algorithm for this setting and showed that it can be used for the design of a truthful mechanism in the BMUD setting with more general feasibility constraints. Alaei et al. [3] studied the prophet matching problem for k-demand and budgeted buyers, in a setting that is equivalent to 1-sided vertex arrival model with known arrival order. They provide $1-\frac{1}{\sqrt{k+3}}$ competitive ratio for k-demand buyers, and $1-\frac{1}{\sqrt{2\pi k}}$ competitive ratio for budgeted buyers with values bounded by $1/k$ of the budget. For the case of $k=1$, they achieve the same $\frac{1}{2}$ competitive ratio as ours, albeit in a more restricted setting, and using a different algorithm.

A recent line of work has considered sample based variants of prophet inequalities, where the distributions of the values are not given explicitly, and the challenge is to provide good competitive ratios using a limited number of samples [6, 12, 13, 20, 46]. Another related line of work, initiated in Kennedy [34, 35], Kertz [36], has considered multiple-choice prophet inequalities, and was later extended to combinatorial settings such as matroid (and matroids intersection) (Kleinberg and Weinberg [37], Azar et al. [6], polymatroids Dütting and Kleinberg [17]), and general downward closed feasibility constrains (Rubinstein [45]).

Online/stochastic matching There is an extensive literature regarding online matching and stochastic matching problems. Below we survey the studies that are most related to our work. Our vertex arrival model restricted to the case of bipartite graphs captures the two-sided vertex arrival model studied by [48, 25], but within a Bayesian setting. For the online setting, Wang and Wong [48] provided a 0.526-competitive fractional algorithm and Gamlath et al. [25] provided a $\frac{1}{2}+\Omega(1)$-competitive integral algorithm. Furthermore, Gamlath et al. [25] showed that for the case of bipartite graphs with edge arrivals, no online algorithm performs better than the straightforward greedy algorithm, which is $\frac{1}{2}$-competitive. Another extension of the 1-sided bipartite matching model, the fully online matching model has been studied by Ashlagi et al. [5], Huang et al. [30, 31, 32], motivated by ride sharing applications. This is a different vertex arrival model in which all vertices from a general graph arrive and depart online. It is possible to study the stochastic/prophet inequality version of the fully online model, which we leave as an interesting future direction. Lee and Singla [40] proposed the batch-arrival model for online matching and designed competitive algorithms that beat the naïve greedy algorithm when there are constant number of batches.

Gravin et al. [26] studied the online stochastic matching problem with edge arrivals (a.k.a. the unweighted version of the prophet inequality with edge arrivals in this paper) and achieved a 0.506-competitive algorithm. The stochastic matching setting is also studied in the (offline) query-commit framework. The input of this problem is an (unweighted) graph associated with the existence probabilities of all edges. The algorithm can query the existence of the edges in any order. However, if an edge exists, it has to be included into the solution. The Ranking algorithm by Karp et al. [33]

induces an $(1-\frac{1}{e})$-competitive algorithm for this problem on bipartite graphs. Costello et al. [14] provided a 0.573- competitive algorithm on general graphs and proved a hardness of 0.898. Gamlath et al. [24] provided a $1-\frac{1}{e}$-competitive algorithm for the weighted version of this problem.

Online Contention Resolution Schemes (OCRS). Online contention resolution schemes have also been studied in settings beyond worst case arrivals. Adamczyk and Wlodarczyk [1] considered the random order model and constructed $\frac{1}{k+1}$-OCRS for intersections of k matroids. Lee and Singla [41] constructed optimal $\frac{1}{2}$-OCRS and $(1-\frac{1}{e})$-OCRS for matroids with arbitrary order and random order, respectively. Offline contention resolution schemes for matching have also attracted attention due to its applications in submodular maximization problems [11, 22, 7], and the connection between the correlation gap and contention resolution schemes [28]. We refer the interested readers to Bruggmann and Zenklusen [7] for a comprehensive recent survey on the topic.

1.4. Paper Roadmap In Section 2 we extend the OCRS and prophet inequality frameworks to settings where elements arrive online in batches. We begin by introducing the general setting of batched arrival. In Section 2.1 we extend the notion of OCRS to batched-OCRS. In Section 2.2 we extend the notion of prophet inequality to batched prophet inequality. In Section 2.3 we establish a reduction from batched prophet inequality to batched OCRS. In Sections 3 and 4 we construct OCRSs for graph matching under the vertex- and edge-arrival models, respectively. Upper bounds on the competitive ratios for the prophet inequality with edge arrivals are provided in Appendix D. Section 5 concludes this paper with a list of open problems and future directions.

2. Model and Preliminaries: Batched Settings Let E be a set of elements, and let $\mathcal{M}$ be a downward closed family of feasible subsets of E, i.e., if $S\in\mathcal{M}$, then $S'\in\mathcal{M}$ for any $S'\subseteq S$. The elements in E are partitioned into T disjoint sets (batches) $B_1,\ldots,B_T$ that arrive online in the order from batch B_1 to batch B_T. I.e., at time t, all elements of batch B_t appear simultaneously. The partition of elements into the batches and their arrival order $(B_t)_{t\in[T]}$ should conform to a certain structure formally specified by a family of all feasible ordered partitions $\mathcal{B}$ of E.

Some examples of feasible ordered partitions include the following: (i) all batches in $\mathcal{B}$ are required to be singletons, (ii) given a partial order π on E, a feasible ordered partition $(B_t)_{t\in[T]}\in\mathcal{B}$ is required to have $\pi(e_t)\leqslant\pi(e_s)$ for any $e_t\in B_t, e_s\in B_s$ where $t<s$, (iii) suppose the set of elements E consists of the edges of a bipartite graph $G=(L,R;E)$, and each batch B_t must contain all edges incident to a vertex $u\in L$.

We illustrate the main concepts in our setting with the following running example.

EXAMPLE 1 (TABULAR FEASIBILITY). In a tabular feasibility setting, there exists a table with n rows and m columns, and the set of elements is $E=\{(i,j)\mid i\in[n], j\in[m]\}$. The collection of feasible sets is any set of elements that contains at most one element from each row and at most one element from each column. The elements arrive in $T=n$ batches. For $t=1,\ldots,n$, batch t contains all elements in row t; i.e., $B_t=\{(t,j)\mid j\in[m]\}$. The algorithm does not know the arrival order of $(B_t)_{t\in[T]}$. The partition family $\mathcal{B}$ of E can be described as a collection of rows $(\{(i,j)\mid j\in[m]\})_{i\in[T]}$. Note that the structure of batches in this example is equivalent to the structure of batches in example (iii) above of 1-sided vertex arrival in complete bipartite graphs, and the feasibility constraints correspond to bipartite graph matching.

2.1. Batched OCRS For a given family of feasible batches $\mathcal{B}$, consider a *sampling scheme* that selects a random subset $R\subseteq E$ as follows: at time t, all elements of batch B_t arrive, of which a random subset $R_t\subseteq B_t$ is *realized*. The realized sets $R_1,\ldots,R_T$ are mutually independent. R is then defined as the random set $R\overset{\text{def}}{=\!=}\bigsqcup_{t\in[T]}R_t$.

Feldman et al. [23] introduce the notion of c-selectable *online contention resolution scheme* (OCRS), as an online selection process that selects a feasible subset of E such that every realized element $e \in R$ is selected with probability at least c, for the special case where every batch is a singleton. We extend the definition of Feldman et al. [23] to batched OCRSs as follows.

Definition 1 (c-selectable batched OCRS). An online selection algorithm ALG is a batched OCRS with respect to a sampling scheme R if it selects a set $I_t \subseteq R_t$ at every time t such that $I \stackrel{\text{def}}{=\!=} \bigsqcup_{t \in [T]} I_t$ is feasible (i.e., $I \in \mathcal{M}$). It is called a c-selectable batched OCRS (or in short c-batched-OCRS) if for every $t \in [T]$, every realization $S \subseteq B_t$ of R_t, and every element $e \in S$, it holds that:

$$\Pr_I \left[e \in I_t \;\middle|\; R_t = S \right] \geqslant c. \tag{1}$$

The algorithm ALG does not know the complete partition into batches and the arrival order of future batches. It only knows the general structure $\mathcal{B}$. Thus, at time t, ALG chooses I_t based on $B_1, \ldots, B_t$, and $R_1, \ldots, R_t$.

Let us demonstrate the concepts above using our running example of tabular feasibility (Example 1). Consider a sampling scheme R that selects one element from each batch uniformly at random. The following is a 1/2-batched OCRS with respect to R (for the case $n \leqslant m \cdot \ln 4$): given the element sampled by R_t in round t, if it is feasible (given previously chosen elements), then we choose it with probability $p_t = \frac{1}{2(1-\frac{1}{2m})^{t-1}}$ ($p_t \leqslant 1$ by the fact that $n \leqslant m \cdot \ln 4$). We claim that for every t, the element sampled by R_t is chosen with probability 1/2. We prove this by induction on t. For $t = 1$, any element in row 1 can be added to I and $p_1 = 1/2$. For $t > 1$, any element in row t can be added to I with probability $(1 - \frac{1}{2m})^{t-1}$. Multiplying this probability by p_t gives exactly 1/2. Thus, this is a 1/2-batched OCRS.

2.2. Batched Prophet Inequality In batched prophet inequality, every element $e \in E$ has a weight w_e. Let $\mathbf{w}^t \stackrel{\text{def}}{=\!=} (w_e)_{e \in B_t}$, and $\mathbf{w} \stackrel{\text{def}}{=\!=} (\mathbf{w}^t)_{t \in [T]}$. Weights are unknown apriori, but for every t, $\mathbf{w}^t$ is independently drawn from a known (possibly correlated) distribution F_t, and $\mathbf{w} \sim \mathbf{F} \stackrel{\text{def}}{=\!=} \prod_{t \in [T]} F_t$; I.e., we allow dependency within batches, but not across batches. Let $\mathbf{w}(S) = \sum_{e \in S} w_e$ for any set $S \subseteq E$. As standard, let $\mathbf{F}_{-t} = \prod_{i \neq t} F_i$. The particular partition of elements into batches and their order are a priori unknown[3], except, of course, that $(B_t)_{t \in [T]}$ must conform to the general structure of $(B_t)_{t \in [T]} \in \mathcal{B}$. All elements of a batch B_t and their weights $\mathbf{w}^t$ are revealed to the algorithm at time t. We assume that the arrival order of the batches is decided by an oblivious adversary, i.e., the adversary can select an arbitrary partition and order of arrival of the batches in $\mathcal{B}$, but does not see the realization of the weights w_e and the algorithm's decisions[4]. Let OPT be a function that given weights $\mathbf{w}$ returns a feasible set of maximum weight (i.e., $\mathsf{OPT}(\mathbf{w}) \in \arg\max_{S \in \mathcal{M}} \mathbf{w}(S)$)[5].

Definition 2 (c-batched-prophet inequality). A batched-prophet inequality algorithm ALG is an online selection process that selects at time t a set $I_t \subseteq B_t$ such that $I \stackrel{\text{def}}{=\!=} \bigsqcup_{t \in [T]} I_t$ is feasible (i.e., $I \in \mathcal{M}$). We say that ALG has competitive ratio c if

$$\mathop{\mathbf{E}}_{\mathbf{w}, I} [\mathbf{w}(I)] \geqslant c \cdot \mathop{\mathbf{E}}_{\mathbf{w}} [\mathbf{w}(\mathsf{OPT}(\mathbf{w}))].$$

[3] Note that $\mathbf{F}$ might impose some constraints on the partition into batches: elements whose weights are dependent must belong to the same batch. No constraint is imposed on elements whose weights are independent and on the order of batches.

[4] The oblivious adversary is a standard assumption in the literature on online algorithms in stochastic settings.

[5] We assume that OPT is deterministic (if a given weight vector $\mathbf{w}$ induces multiple feasible sets of maximal weight, $\mathsf{OPT}(\mathbf{w})$ returns one of them consistently).

The batched prophet inequality setting corresponding to Example 1 is one where the elements arrive in the same batch structure, but also have real-valued weights that are drawn from known distributions. Say, in our running example $w_{(i,j)} = \xi_i \cdot \xi_{i,j}$, where $\xi_i \sim$ Uniform$[0,1]$ and $\xi_{i,j} \sim$ Uniform$[0,100]$. The weights $w_{(i_1,j_1)}$ and $w_{(i_2,j_2)}$ are independent random variables for any $i_1 \neq i_2$.

2.3. Reduction: Prophet Inequality to OCRS We define a random sampling scheme $R(\mathbf{w},\mathbf{F})$ for $\mathbf{w} \sim \mathbf{F}$ as follows:

Definition 3 (Sampler). Let $R_t(\mathbf{w}^t,\mathbf{F}) \stackrel{\text{def}}{=\!=} B_t \cap \mathsf{OPT}(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)})$ be the random subset of B_t where $\widetilde{\mathbf{w}}^{(t)} \sim \mathbf{F}_{-t}$ are generated independently of $\mathbf{w}$, and $R(\mathbf{w},\mathbf{F}) \stackrel{\text{def}}{=\!=} \bigsqcup_{t\in[T]} R_t(\mathbf{w}^t,\mathbf{F})$.

Note that:
1. The distribution of $R(\mathbf{w},\mathbf{F})$ is a product distribution over the random variables $R_t(\mathbf{w}^t,\mathbf{F})$.
2. Since $\mathbf{F}$ is a product distribution, $(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)}) \sim \mathbf{F}$.
3. $\forall t \in [T]$, $R_t(\mathbf{w}^t,\mathbf{F}) = R(\mathbf{w},\mathbf{F}) \cap B_t$ has the same distribution as $\mathsf{OPT}(\mathbf{w}) \cap B_t$, where $\mathbf{w} \sim \mathbf{F}$.
4. $\forall t \in [T]$, $R_t(\mathbf{w}^t,\mathbf{F}) \in \mathcal{M}$, and $\mathsf{OPT}(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)}) \in \mathcal{M}$. But, $R(\mathbf{w},\mathbf{F})$ might not belong to $\mathcal{M}$.
5. For every $t \in [T]$,

$$\mathop{\mathbf{E}}_{\mathbf{w},R}\left[\mathbf{w}(R_t(\mathbf{w}^t,\mathbf{F}))\right] = \mathop{\mathbf{E}}_{\mathbf{w}}\left[\mathbf{w}(\mathsf{OPT}(\mathbf{w}) \cap B_t)\right]. \tag{2}$$

In Example 1, the sampler defined in Definition 3 observes the weights of the realized batch at each time $t \in [T]$, samples each $\xi_i \sim$ Uniform$[0,1]$ and $\xi_{i,j} \sim$ Uniform$[0,100]$ for $i \neq t$ and calculates the weights of all other elements $w_{i,j} = \xi_i \cdot \xi_{i,j}$, where $i \neq t$ and $j \in [m]$. The weights $\mathbf{w}^t$ in each B_t are distributed as $(\xi_t \cdot \xi_{t,j})_{j\in[m]}$. Then, the sampler finds the maximum weight feasible set and selects in it the element (if it exists) from the current batch B_t. For a random $\mathbf{w}^t$, the sampling scheme $R_t(\mathbf{w}^t,\mathbf{F})$ picks a single element $(t,j) \in B_t$ with probability $\min(\frac{m}{n},1)$ where $j \sim$ Uniform$[m]$. Thus we select in $R(\mathbf{w},\mathbf{F})$ one element from each row t independently and uniformly at random with probability $\min(\frac{m}{n},1)$. It is easy to see that $R(\mathbf{w},\mathbf{F})$ satisfies all properties 1-5 from above.

Theorem 1 **(reduction: c-batched prophet inequality to c-batched OCRS)**. *For every set $\mathcal{B}$ of feasible ordered partitions, given a c-batched OCRS for the sampling scheme $R(\mathbf{w},\mathbf{F})$ with $\mathbf{w} \sim \mathbf{F}$, there is a batched prophet inequality algorithm for $\mathbf{w} \sim \mathbf{F}$ with competitive ratio c.*

Proof. Consider the following online algorithm:

Algorithm 1 Reduction from c-batched prophet inequality to c-batched OCRS

1: **for** $t \in \{1,...,T\}$ **do**
2: Let $\mathbf{w}^t$ be the weights of elements in B_t
3: Resample the weights $\widetilde{\mathbf{w}}^{(t)} \sim \mathbf{F}_{-t}$
4: Let $R_t \leftarrow \mathsf{OPT}(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)}) \cap B_t$
5: $I_t \leftarrow c$-OCRS$(B_1,\ldots,B_t,R_1,\ldots,R_t)$ for the structure $\mathcal{B}$ of batches.
6: **end for**
7: Return $I = \bigsqcup_{t\in[T]} I_t$

Let I be the random set returned by Algorithm 1, and R_t (and resp. $R = \bigsqcup_{t\in[T]} R_t$) be the sets defined in step 4 of the algorithm. It holds that

$$\mathop{\mathbf{E}}_{\mathbf{w},R,I}[\mathbf{w}(I)] = \sum_{t\in[T]} \mathop{\mathbf{E}}_{\mathbf{w},R,I}[\mathbf{w}(I_t)] = \sum_{t\in[T]} \sum_{S\subseteq B_t} \mathop{\mathbf{E}}_{\mathbf{w},R,I}\left[\mathbf{w}(I_t) \;\middle|\; R_t = S\right] \mathop{\mathbf{Pr}}_{\mathbf{w},R}[R_t = S].$$

Since I_t and $\mathbf{w}$ are independent given that $R_t = S$, we also have.

$$\begin{aligned}
\mathop{\mathbf{E}}_{\mathbf{w},R,I}[\mathbf{w}(I)] &= \sum_{t\in[T]}\sum_{S\subseteq B_t}\mathop{\mathbf{E}}_{\mathbf{w},I}\left[\sum_{e\in B_t} w_e\cdot \mathop{\mathbf{Pr}}_{I}[e\in I_t]\;\middle|\; R_t=S\right]\mathop{\mathbf{Pr}}_{\mathbf{w}^t,R_t}[R_t=S] \\
&\overset{(1)}{\geqslant} \sum_{t\in[T]}\sum_{S\subseteq B_t}\mathop{\mathbf{E}}_{\mathbf{w}^t}\left[\sum_{e\in S} w_e\cdot c\;\middle|\; R_t=S\right]\mathop{\mathbf{Pr}}_{\mathbf{w}^t,R_t}[R_t=S] \\
&= c\sum_{t\in[T]}\sum_{S\subseteq B_t}\mathop{\mathbf{E}}_{\mathbf{w}}\left[\mathbf{w}(S)\;\middle|\; R_t=S\right]\mathop{\mathbf{Pr}}_{\mathbf{w},R_t}[R_t=S] \\
&= c\sum_{t\in[T]}\mathop{\mathbf{E}}_{\mathbf{w},R}[\mathbf{w}(R_t)] \overset{(2)}{=} c\sum_{t\in[T]}\mathop{\mathbf{E}}_{\mathbf{w}}[\mathbf{w}(\mathsf{OPT}(\mathbf{w})\cap B_t)] = c\cdot\mathop{\mathbf{E}}_{\mathbf{w}}[\mathbf{w}(\mathsf{OPT}(\mathbf{w}))] \qquad \square
\end{aligned}$$

We note that our reduction is similar to the classical case in which elements arrive one by one. However, the set of requirements from the sampling scheme is more demanding in the batched setting. In particular, our sampling scheme needs to satisfy all 5 properties mentioned right after Definition 3.

The difference between the batched setting and the standard singleton setting can be best demonstrated when using the ex-ante relaxation benchmark instead of the integral optimal solution (OPT) benchmark. In the ex-ante relaxation case the sampling scheme in the batched setting would fail to satisfy Property 4, as the ex-ante relaxation may select more than a single element within a batch (whereas Property 4 requires the sampler to select a feasible set within each batch). Clearly, under singleton arrivals, Property 4 is trivially satisfied also in the ex-ante relaxation case.

3. A 1/2-Batched OCRS for Matching with Vertex Arrival Given a graph $G = (V, E)$ (not necessarily bipartite), the elements of the prophet inequality setting are the edges $e \in E$, and the family of feasible sets $\mathcal{M}$ is given by all matchings in G, i.e., $M \subseteq E$ is feasible iff $e_1 \cap e_2 = \varnothing$ for any $e_1, e_2 \in M$.

In the vertex arrival model, the vertices arrive in an arbitrary unknown order σ: $v_{\sigma(1)}, \ldots, v_{\sigma(n)}$, where $v_{\sigma(i)}$ is the vertex arriving at time i. Upon arrival of vertex $v_{\sigma(i)}$, the weights on the edges from $v_{\sigma(i)}$ to all previous vertices $v_{\sigma(j)}$, where $j < i$, are revealed to the algorithm. The online algorithm must make an immediate and irrevocable decision whether to match $v_{\sigma(i)}$ to some available vertex $v_{\sigma(j)}$ such that $j < i$ (in which case $v_{\sigma(i)}$ and $v_{\sigma(j)}$ become unavailable), or leave $v_{\sigma(i)}$ unmatched (in which case $v_{\sigma(i)}$ remains available for future matches). Let $B_i^\sigma \overset{\text{def}}{=} \{(v_{\sigma(i)}v_{\sigma(j)}) \mid j < i\}$. The set of feasible ordered partitions for the vertex arrival model is

$$\mathcal{B}^{v.a.} \overset{\text{def}}{=} \{(B_1^\sigma, \ldots, B_{|V|}^\sigma)\}_{\sigma\in S_V},$$

where S_V is the set of permutations over V.

In what follows we construct a $\frac{1}{2}$-batched OCRS for the vertex arrival model. By the reduction in Theorem 1, the constructed batched OCRS gives a batched prophet inequality with competitive ratio 1/2 with respect to the optimal matching. The sampling scheme R (see Definition 3) for the vertex arrival batch structure is as follows.

Definition 4 (Sampler). For every vertex v arriving at time t, let R_v be an independent random subset of B_v^σ generated by the sampling scheme ($R_v = B_v^\sigma \cap \mathsf{OPT}(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)})$, where $\mathbf{w}^t$ are the observed weights in the batch B_v^σ and $\widetilde{\mathbf{w}}^{(t)} \sim \mathbf{F}_{-t}$ are generated independently of $\mathbf{w}$), and let $R = \bigsqcup_v R_v$.

For every edge $(uv) \in E$, let $x_{uv} \overset{\text{def}}{=} \mathbf{Pr}[(uv) \in R]$.We write $u < v$ if vertex u arrives before vertex v in the vertex arrival order σ. In order to construct a $\frac{1}{2}$-batched OCRS for R, we first claim that R satisfies the following two equations:

$$\sum_u x_{uv} \leqslant 1 \quad \text{for every } v \in V \tag{3}$$

$$|R_v| \leqslant 1 \quad \text{for every } v \in V \text{ and every realization of } R_v \tag{4}$$

R satisfies (3) since the probability that vertex v is matched in a maximum matching is at most 1. It also satisfies (4), since the sampler matches v to at most one vertex upon the arrival of vertex v (i.e., selects at most one element).

With this, we are ready to present our main theorem for this section.

THEOREM 2. *R admits a 1/2-batched OCRS for the $\mathcal{B}^{v.a.}$ structure of batches.*

Proof. Upon the arrival of a vertex v, we compute $\alpha_u(v)$ for every $u < v$ as follows:

$$\alpha_u(v) \overset{\text{def}}{=\!=} \frac{1}{2 - \sum_{z<v} x_{uz}} \leqslant \frac{1}{2 - \sum_z x_{uz}} \overset{(3)}{\leqslant} 1. \tag{5}$$

Note that $\alpha_u(v)$ cannot be calculated before the arrival of v. We claim that the following algorithm is a $\frac{1}{2}$-batched OCRS with respect to R:

Algorithm 2 1/2-batched OCRS for vertex arrival

1: **for** $v \in \{1, ..., |V|\}$ **do**
2: Calculate $x_{uz} = \mathbf{Pr}[(uz) \in R]$ for all $u, z < v$ and $\alpha_u(v)$ for all $u < v$.
3: Match the edge $(uv) \in R_v$ (if $R_v \neq \varnothing$) with probability $\alpha_u(v)$ if u is unmatched.
4: **end for**

Note that Algorithm 2 is well defined, since by Equation (5), $\alpha_u(v) \leqslant 1$ and Algorithm 2 matches no more than one vertex to v by (4). It remains to show that Algorithm 2 is a 1/2-batched OCRS with respect to R. We prove that $\mathbf{Pr}[(uv) \text{ is matched}] = \frac{x_{uv}}{2}$ by induction on the vertices V, according to the arrival order. For the base case, $V = \varnothing$, the argument trivially holds. For the induction step, assume that $\mathbf{Pr}[(uz) \text{ is matched}] = \frac{x_{uz}}{2}$ for all $u, z < v$. We show that $\mathbf{Pr}[(uv) \text{ is matched}] = \frac{x_{uv}}{2}$ for all $u < v$. In what follows, we say that "u is unmatched *at* v" if u is unmatched *right before* v arrives.

$$\mathbf{Pr}[u \text{ is unmatched at } v] = 1 - \sum_{z<v} \mathbf{Pr}[(uz) \text{ is matched}] = 1 - \frac{1}{2}\sum_{z<v} x_{uz}, \tag{6}$$

where the second equality follows from the induction hypothesis. Therefore,

$$\begin{aligned} \mathbf{Pr}[(uv) \text{ is matched}] &= \mathbf{Pr}[u \text{ is unmatched at } v] \cdot \mathbf{Pr}[(uv) \in R_v] \cdot \alpha_u(v) \\ &\overset{(5),(6)}{=} \left(1 - \frac{1}{2}\sum_{z<v} x_{uz}\right) \cdot \frac{1}{2 - \sum_{z<v} x_{uz}} \cdot x_{uv} = \frac{x_{uv}}{2}. \end{aligned}$$

In order to prove that Algorithm 2 is a $\frac{1}{2}$-batched OCRS with respect to R, we need to show that $\mathbf{Pr}[(uv) \in I_v \mid R_v = \{(uv)\}] \geqslant 1/2$ for every $u < v$. Indeed,

$$\begin{aligned} \mathbf{Pr}[(uv) \in I_v \mid R_v = \{(uv)\}] &= \mathbf{Pr}[u \text{ is unmatched at } v] \cdot \alpha_u(v) \\ &\overset{(5),(6)}{=} \left(1 - \frac{1}{2}\sum_{z<v} x_{uz}\right) \cdot \frac{1}{2 - \sum_{z<v} x_{uz}} = \frac{1}{2}. \end{aligned}$$ □

Computational aspects. Here we discuss how our algorithm can be implemented efficiently. Note that given the probabilities $\{x_{uv}\}_{(uv)\in E}$, we can calculate $\{\alpha_u(v)\}_{(uv)\in E}$ by Equation (5). Thus our batched OCRS can be implemented in polynomial time, if we are explicitly given the probability density functions of each element in R. However, the batched OCRS in the reduction to prophet inequality, grants us only sample access to R. In a sense, it is a problem of calculating the value and estimating basic statistics of the maximum weighted matching benchmark. If we have only a sample access to R, we can still apply standard Monte-Carlo algorithm to estimate x_{uv}'s within arbitrary additive accuracy (with high probability), which leads to estimation of $\{\alpha_u(v)\}_{(uv)\in E}$ within arbitrary multiplicative accuracy (by Equation (5) and the fact that $\alpha_u(v)\geqslant \frac{1}{2}$). This gives us a $(\frac{1}{2}-\epsilon)$-batched OCRS that runs in poly$(|V|,\frac{1}{\epsilon})$ time.

Guarantees against a stronger benchmark. The guarantee in Definition 2 can be strengthen to hold against the stronger benchmark of the optimal fractional matching. This extension of the definition of the batched OCRS and the reduction from batched prophet inequality to batched OCRS are presented in Appendix A. The construction of the OCRS for the setting of fractional matching in the vertex arrival model is given in Appendix C.

4. A 0.337-OCRS for Matching with Edge Arrival Given a graph $G=(V,E)$ (not necessarily bipartite), the elements of the prophet inequality setting are the edges $e\in E$, and the family of feasible sets $\mathcal{M}$ is given by all matchings in G, i.e., $M\subseteq E$ is feasible $M\in\mathcal{M}$ iff $e_1\cap e_2=\varnothing$ for any $e_1,e_2\in M$.

In the edge arrival model, the edges arrive in an arbitrary unknown order σ: $e_{\sigma(1)},\dots,e_{\sigma(|E|)}$. Upon arrival of edge $e=(uv)$, the algorithm must decide whether to match it (provided that u and v are still unmatched), or leave e unmatched potentially saving u and/or v for future matches. Let B_i^σ be the singleton $\{e_{\sigma(i)}\}$. The set of feasible ordered partitions for the edge arrival model is

$$\mathcal{B}^{e.a.} \stackrel{\text{def}}{=\!=} \{(B_1^\sigma,\dots,B_{|E|}^\sigma)\}_{\sigma\in S_E},$$

where S_E is the set of permutations over E.

In what follows we construct a c-OCRS for the edge arrival model. We start with a warm-up in Section 4.1, establishing a $\frac{1}{3}$-OCRS. In Section 4.2 we present an improved 0.337-OCRS, using subtle observations about correlated events. Our results imply a prophet inequality with competitive ratio 0.337 for the edge-arrival model. In Appendix B we show that this guarantee holds also with respect to the optimal *ex-ante* matching (which is a stronger benchmark; stronger even than the optimal *fractional* matching).

In batched OCRS we define a sampling scheme R (independent across batches), which in turn defines corresponding marginals $\mathbf{x}$. If every batch consists of a single element (as in the model of matching with edge arrival), any vector of marginal probabilities $\mathbf{x}\in[0,1]^E$ induces the unique sampling scheme R. Hence, R is described by $\mathbf{x}\in[0,1]^E$. Let $\mathbf{x}$ be any probability vector such that

$$\sum_{e:v\in e} x_e \leqslant 1 \text{ for all } v\in V. \tag{7}$$

Note that the sampling scheme R defined in Definition 3, when applied to matching with edge arrival, satisfies this condition.

Let σ be an arbitrary (unknown) order of the edges. Let $R=\bigsqcup_{e\in E} R_{\sigma(e)}$ be a sampling scheme that independently generates R_e for each edge e as follows. $R_e=\{e\}$ with probability x_e, and $R_e=\varnothing$ otherwise. To simplify notation, we sometimes use x_{uv} to denote $x_{(uv)}$. Recall that the definition of c-OCRS requires the selected set I to be feasible, and each element $e\in E$ to satisfy $\mathbf{Pr}[e\in I\mid e\in R_e]\geqslant c$. That is, the probability that e is selected given that it is in R_e should be at least c. Our algorithm will actually guarantee the last inequality with equality, namely that $\mathbf{Pr}[e\in I\mid e\in R_e]=c$

Algorithm 3 c-OCRS for edge arrival

1: At the arrival of edge (uv)
2: Given the arrival order $\sigma_{<(uv)}$ of the edges preceding (uv), calculate $\mathbf{Pr}[u, v \text{ are unmatched at } (uv)]$.
3: Define
$$\alpha_{(uv)} \stackrel{\text{def}}{=\!=} \frac{c}{\mathbf{Pr}[u, v \text{ are unmatched at } (uv)]} \tag{8}$$
4: If (i) u, v are unmatched, and (ii) $(uv) \in R_{(uv)}$, then match (uv) with probability $\alpha_{(uv)}$.

for all $e \in E$. In the description of the algorithm and throughout this section, we write "*at* (uv)" or "*at* e" as a shorthand notation to indicate the time right *before* the arrival of the edge $e = (uv)$.

Note that the term $\mathbf{Pr}[u, v \text{ are unmatched at } (uv)]$ involves both randomness from R and from previous steps of our algorithm. It holds that:

$$\mathbf{Pr}\left[(uv) \text{ is matched} \mid (uv) \in R_{(uv)}\right] = \mathbf{Pr}\left[u, v \text{ are unmatched at } (uv)\right] \cdot \alpha_{(uv)} \stackrel{(8)}{=} c, \tag{9}$$

which satisfies the inequality required by c-OCRS (Equation (1)).

It remains to show that Algorithm 3 is well-defined, i.e., that $\alpha_e \leqslant 1$ for all $e \in E$. In Section 4.1 we show that $c = \frac{1}{3}$ can be proved using a relatively simple analysis. In Section 4.2 we present a more involved analysis showing that one can improve 1/3 to $c = 0.337$.

Computational aspects. The computation of $\{x_e\}_{e\in E}$ is similar to the vertex arrival setting. In fact, we can work with a stronger benchmark of the ex-ante relaxation in the edge arrival setting, which is easier from the computational view point and admits a polynomial time algorithm that finds $\{x_e\}_{e\in E}$ as the solution to the ex-ante relaxation. By contrast to the vertex arrival setting, given $\{x_{uv}\}_{(uv)\in E}$, it might take exponential time to precisely calculate $\{\alpha_{(uv)}\}_{(uv)\in E}$ in Algorithm 3. We still can use Monte-Carlo method to estimate $\alpha_{(uv)}$'s within arbitrary multiplicative accuracy (by the fact that $\alpha_{(uv)} \geqslant \frac{1}{3}$), which results in a $(0.337 - \epsilon)$-OCRS that runs in $\text{poly}(|V|, \frac{1}{\epsilon})$ time.

Guarantees against a stronger benchmark. In Appendix B we show that for the edge arrival model, our construction gives an approximation with respect to an even stronger benchmark, known as the ex-ante relaxation.

4.1. Warm-up: $\frac{1}{3}$-OCRS

THEOREM 3. *There is a $\frac{1}{3}$-OCRS for matching in general graphs with edge arrivals.*

Proof. Let $c = \frac{1}{3}$. We prove that all $\alpha_e \leqslant 1$ for every edge e by induction on the set of edges, according to their arrival order. For the base case (an empty set), the argument holds trivially. We next prove the induction step. We can assume by the induction hypothesis that $\alpha_e \leqslant 1$ for every edge $e \in E$ but the last arriving edge (uv). To finish the induction step we need to show that $\alpha_{(uv)} \leqslant 1$. Recall that our algorithm matches each edge e preceding (uv) with probability $c \cdot x_e$. Therefore,

$$\mathbf{Pr}\left[u \text{ is matched at } (uv)\right] = \sum_{s\neq v} c \cdot x_{us} \leqslant c \quad \text{and} \quad \mathbf{Pr}\left[v \text{ is matched at } (uv)\right] = \sum_{s\neq u} c \cdot x_{sv} \leqslant c. \tag{10}$$

Indeed, the events that u is matched to the vertex s for each $s \in V \backslash \{v\}$ are disjoint, $\mathbf{Pr}[u \text{ matched to } s] = c \cdot x_{us}$, and $\sum_s x_{us} \leqslant 1$; similar argument applies to $\mathbf{Pr}[v \text{ is matched at } (uv)]$. By the union bound, we have

$$\mathbf{Pr}\left[u, v \text{ are unmatched at } (uv)\right] \geqslant 1 - \mathbf{Pr}\left[u \text{ is matched at } (uv)\right] - \mathbf{Pr}\left[v \text{ is matched at } (uv)\right] \geqslant 1 - 2c.$$

For $c = 1/3$, $1 - 2c = c$. Thus,

$$\mathbf{Pr}\left[u,v \text{ are unmatched at } (uv)\right] \geqslant c \quad \text{and} \quad \alpha_{(uv)} = \frac{c}{\mathbf{Pr}[u,v \text{ are unmatched at } (uv)]} \leqslant 1,$$

as desired. This concludes the proof. ◻

4.2. Improved Analysis: 0.337-OCRS In order to improve the competitive ratio beyond 1/3, we strengthen the lower bound on the probability that u, v are unmatched at (uv). We again apply the same inductive argument as in the warm-up, but use more complex estimate on $\mathbf{Pr}[u,v \text{ are unmatched at } (uv)]$ than a simple union bound. We denote

$$x_u \stackrel{\text{def}}{=\!=} \sum_{s \notin \{u,v\}} x_{us} \leqslant 1 \quad \text{and} \quad x_v \stackrel{\text{def}}{=\!=} \sum_{s \notin \{u,v\}} x_{sv} \leqslant 1.$$

Similar to (10) we have

$$\mathbf{Pr}\left[u \text{ is matched at } (uv)\right] \leqslant c \cdot x_u \quad \text{and} \quad \mathbf{Pr}\left[v \text{ is matched at } (uv)\right] \leqslant c \cdot x_v. \tag{11}$$

Hence, by the inclusion-exclusion principle we have

$$\begin{aligned} &\mathbf{Pr}\left[u,v \text{ are unmatched at } (uv)\right] \\ &= 1 - \mathbf{Pr}\left[u \text{ is matched at } (uv)\right] - \mathbf{Pr}\left[v \text{ is matched at } (uv)\right] + \mathbf{Pr}\left[u,v \text{ are matched at } (uv)\right] \\ &\geqslant 1 - c\cdot(x_u + x_v) + \mathbf{Pr}\left[u,v \text{ are matched at } (uv)\right]. \end{aligned} \tag{12}$$

If the matching statuses of u and v were independent, the bound (12) would be $1 - c(x_u + x_v) + c^2 x_u x_v \geqslant 1 - 2c + c^2$, and equating it to c would yield $c \approx 0.382$. However, it is possible that the events that u and v are matched are negatively correlated. The following lemma gives a non-trivial lower bound on this correlation. This is the most technical lemma in this paper; its proof is the content of Section 4.3.

Lemma 1. *For every $c \in [0, \frac{1}{2}]$*

$$\mathbf{Pr}\left[u,v \textit{ are umatched at } (uv)\right] \geqslant 1 - 2c + \frac{c^2}{2} \cdot \left(\frac{1-2c}{1-c}\right)^2.$$

The bound in Lemma 1 leads to the construction of the improved 0.337-OCRS.

Theorem 4. *There is a 0.337-OCRS for general graphs with edge arrivals.*

Proof. We set $c \approx 0.337$ to be the solution of $1 - 2c + \frac{c^2}{2} \cdot \left(\frac{1-2c}{1-c}\right)^2 = c$. Then by Lemma 1 $\mathbf{Pr}[u,v \text{ are unmatched at } (uv)] \geqslant c$ and $\alpha_{(uv)} = \frac{c}{\mathbf{Pr}[u,v \text{ are unmatched at } (uv)]} \leqslant 1$, as required. ◻

4.3. Proof of Lemma 1 Fix an edge arrival order σ. We prove Lemma 1 by induction on the set of edges, according to the arrival order σ. The base case (the empty set) holds trivially. We next prove the induction step. By the induction hypothesis, we can assume that $\alpha_e \leqslant 1$ for every edge $e \in E$ but the last edge (uv) in σ. To simplify notations, we slightly abuse the definition of E by excluding edge (uv) from E. We need to show that $\alpha_{(uv)} \leqslant 1$. By the induction hypothesis, Algorithm 3 matches each edge $e \in E$ with probability exactly $c \cdot x_e$. For the purpose of analysis, we think of the following random procedure that unifies the random realization in R and the random decisions made by our algorithm.

1. For each $e \in E$, $e \in R_e$ with probability x_e, and conditioned on the event $e \in R_e$, e is active with probability α_e.

2. Greedily pick active edges according to the arrival order σ. I.e., pick an active edge (uv) if both u and v are unmatched at (uv).

In the above procedure, each edge e is *active* with probability $\alpha_e \cdot x_e$ (independently across edges). Then, it is matched if both its ends are unmatched at the time the edge arrives. In the remainder of this section we give a lower bound on the probability that both u, v are unmatched at (uv).

Let $\widetilde{E} \subseteq E$ be the set of the active edges. Suppose there is a vertex u^* such that $(uu^*) \in \widetilde{E}$ is the only active edge of u^*. Then u^* must remain unmatched before (uu^*). When (uu^*) arrives, u is either matched before, or it will be matched now. We call such u^* a *witness* of u, as existence of u^* implies that u is matched. Moreover, if both u and v admit witnesses u^*, v^*, then u, v must be matched at (uv). Note that by definition $u^* \neq v^*$.

Let us give a lower bound on the probability that each of u, v have a witness. We first describe a sampler π_u that given the set of active edges $\widetilde{E}_u \subseteq E_u \stackrel{\text{def}}{=\!=} \{e \in E | e \text{ incident to } u\}$ incident to u, proposes a candidate witness of u. Let $\pi_u : 2^E \to V \cup \{\text{null}\}$ be the following random mapping.

1. Resample each $e \in E_{\text{-}u} \stackrel{\text{def}}{=\!=} E \backslash E_u$ independently with probability $\alpha_e \cdot x_e$. Let the active edges be $\widehat{E}_{\text{-}u} \subseteq E_{\text{-}u}$.
2. Run greedy on the instance $G = (V, \widetilde{E}_u \cup \widehat{E}_{\text{-}u})$ according to the arrival order σ.
3. If u is matched with a vertex s, return $\pi_u(\widetilde{E}_u) = s$; else, return null.

The sampling procedure corresponds to the actual run of our algorithm, since $\widetilde{E}_u \cup \widehat{E}_{\text{-}u}$ has the same distribution as $\widetilde{E}$. Thus, the probability that u^* is returned as the candidate witness of u equals the probability that (uu^*) is matched by our algorithm, which equals $c \cdot x_{uu^*}$. Hereafter, we denote the event that a vertex u^* is chosen by the sampler π_u as the candidate witness of a vertex u by "u^* candidate of u". Thus

$$\mathop{\mathbf{Pr}}_{\widetilde{E}_u, \pi_u} [u^* \text{ candidate of } u] = c \cdot x_{uu^*}. \tag{13}$$

We also define a similar sampler π_v to generate the candidate witness of v. Then,

$$\begin{aligned}
&\mathbf{Pr}[u, v \text{ have witnesses}] \\
&\quad \geqslant \sum_{\substack{u^* \neq v^* \\ u^*, v^* \notin \{u,v\}}} \mathop{\mathbf{Pr}}_{\widetilde{E}, \pi_u, \pi_v} \left[u^* \text{ candidate of } u, v^* \text{ candidate of } v, |\widetilde{E} \cap E_{u^*}| = |\widetilde{E} \cap E_{v^*}| = 1\right] \\
&\quad = \sum_{\substack{u^* \neq v^* \\ u^*, v^* \notin \{u,v\}}} \mathop{\mathbf{Pr}}_{\widetilde{E}_u, \pi_u} [u^* \text{ candidate of } u] \cdot \mathop{\mathbf{Pr}}_{\widetilde{E}_u, \pi_u} \left[(uv^*) \notin \widetilde{E} \;\middle|\; u^* \text{ candidate of } u\right] \\
&\qquad\qquad \times \mathop{\mathbf{Pr}}_{\widetilde{E}_v, \pi_v} [v^* \text{ candidate of } v] \cdot \mathop{\mathbf{Pr}}_{\widetilde{E}_v, \pi_v} \left[(vu^*) \notin \widetilde{E} \;\middle|\; v^* \text{ candidate of } v\right] \\
&\qquad\qquad \times \mathop{\mathbf{Pr}}_{\widetilde{E}(G - \{u,v\})} [(su^*) \text{ and } (sv^*) \text{ not active } \forall s \in V \backslash \{u, v\}]
\end{aligned} \tag{14}$$

LEMMA 2. *For all $u, i, j \in V$ such that $i \neq j$,*

$$\mathop{\mathbf{Pr}}_{\widetilde{E}_u, \pi_u} \left[(ui) \textit{ is not active} \;\middle|\; j \textit{ candidate of } u\right] \geqslant \mathop{\mathbf{Pr}}_{\widetilde{E}_u} [(ui) \textit{ is not active}]$$

Proof. As $1 - \mathbf{Pr}[(ui) \text{ is active} \mid j \text{ candidate of } u] = \mathbf{Pr}[(ui) \text{ is not active} \mid j \text{ candidate of } u]$ and $1 - \mathbf{Pr}[(ui) \text{ is active}] = \mathbf{Pr}[(ui) \text{ is not active}]$, we just need to show

$$\mathbf{Pr}[(ui) \text{ is active}] \geqslant \mathbf{Pr}\left[(ui) \text{ is active} \;\middle|\; j \text{ candidate of } u\right],$$

which is equivalent to

$$\mathop{\mathbf{Pr}}_{\widetilde{E}_u, \pi_u} \left[j \text{ candidate of } u \;\middle|\; (ui) \text{ is active}\right] \leqslant \mathop{\mathbf{Pr}}_{\widetilde{E}_u, \pi_u} [j \text{ candidate of } u].$$

Two types of randomness are involved in this statement, the realization of edges $\widetilde{E}_u$ that are incident to u and the resampling of remaining edges $\widehat{E}_{\text{-}u}$ in $\pi_u(\widetilde{E}_u)$. Fix the realization of $\widetilde{E}_u \backslash (ui)$ and $\widehat{E}_{\text{-}u}$. If j is chosen as the candidate by π_u when (ui) is active, it must also be chosen when (ui) is not active. This finishes the proof of the lemma. ◻

By Lemma 2, $\underset{\widetilde{E}_u,\pi_u}{\mathbf{Pr}}\left[(uv^*) \text{ is not active } \middle| u^* \text{ candidate of } u\right] \geqslant \underset{\widetilde{E}_u}{\mathbf{Pr}}[(uv^*) \text{ is not active}]$ and $\underset{\widetilde{E}_v,\pi_v}{\mathbf{Pr}}\left[(vu^*) \text{ is not active } \middle| v^* \text{ candidate of } v\right] \geqslant \underset{\widetilde{E}_v}{\mathbf{Pr}}[(vu^*) \text{ is not active}]$ in (14). Furthermore,

$$\underset{\widetilde{E}_u}{\mathbf{Pr}}\left[(uv^*) \notin \widetilde{E}\right] \times \underset{\widetilde{E}_v}{\mathbf{Pr}}\left[(vu^*) \notin \widetilde{E}\right] \times \underset{\widetilde{E}(G-\{u,v\})}{\mathbf{Pr}}\left[\forall s \neq u,v \quad (su^*),(sv^*) \notin \widetilde{E}\right]$$
$$= \prod_{\substack{e=(sv^*)\\ s\neq v}} \underset{\widetilde{E}}{\mathbf{Pr}}\left[e \notin \widetilde{E}\right] \prod_{\substack{e=(su^*)\\ s\neq u}} \underset{\widetilde{E}}{\mathbf{Pr}}\left[e \notin \widetilde{E}\right] \geqslant \underset{\widetilde{E}_{u^*}}{\mathbf{Pr}}\left[(u^*s) \text{ not active } \forall s\right] \cdot \underset{\widetilde{E}_{v^*}}{\mathbf{Pr}}\left[(v^*s) \text{ not active } \forall s\right].$$

We also know that $\mathbf{Pr}[u^* \text{ candidate of } u] = c \cdot x_{uu^*}$ and $\mathbf{Pr}[v^* \text{ candidate of } v] = c \cdot x_{vv^*}$. So we can continue the lower bound (14) on $\mathbf{Pr}[u,v \text{ have witnesses}]$ as follows

$$(14) \geqslant \sum_{\substack{u^*\neq v^*\\ u^*,v^*\notin\{u,v\}}} c^2 \cdot x_{uu^*} x_{vv^*} \underset{\widetilde{E}_{u^*}}{\mathbf{Pr}}\left[(u^*s) \text{ not active } \forall s\right] \cdot \underset{\widetilde{E}_{v^*}}{\mathbf{Pr}}\left[(v^*s) \text{ not active } \forall s\right]$$

Lemma 3. *For any vertex r,* $\underset{\widetilde{E}_r}{\mathbf{Pr}}[(rs) \text{ *is not active* } \forall s] \geqslant \frac{1-2c}{1-c}$.

Proof. Without loss of generality, we assume that neighbors of r are enumerated from 1 to k in such a way that among all edges incident to r, the edge (ri) appears as the i-th edge in σ. Notice that each edge (ri) is active independently with probability $\alpha_{ri}x_{ri}$. Recall that $\alpha_{ri} = c \cdot \mathbf{Pr}_{\widetilde{E}}[r,i \text{ are unmatched at } (ri)]^{-1}$. As r is matched to j with probability $c \cdot x_{rj}$, we have by a union bound and induction hypothesis

$$\underset{\widetilde{E}}{\mathbf{Pr}}\left[r,i \text{ are unmatched at } (ri)\right] \geqslant 1 - \underset{\widetilde{E}}{\mathbf{Pr}}\left[i \text{ is matched at } (ri)\right] - \underset{\widetilde{E}}{\mathbf{Pr}}\left[r \text{ is matched at } (ri)\right]$$
$$\geqslant 1 - c - \sum_{j<i} \underset{\widetilde{E}}{\mathbf{Pr}}\left[(rj) \text{ is matched}\right] = 1 - c - c\sum_{j<i} x_{rj}. \quad (15)$$

Furthermore, each edge (ri) is active independently with probability $\alpha_{ri}x_{ri}$. Therefore, we have

$$\underset{\widetilde{E}}{\mathbf{Pr}}\left[(ri) \text{ is not active } \forall i\right] = \prod_{i=1}^{k}(1-\alpha_{ri}x_{ri}) = \prod_{i=1}^{k}\left(1 - \frac{c \cdot x_{ri}}{\underset{\widetilde{E}}{\mathbf{Pr}}[r,i \text{ are unmatched at } (ri)]}\right)$$
$$\geqslant \prod_{i=1}^{k}\left(1 - \frac{c\cdot x_{ri}}{1-c-c\sum_{j<i}x_{rj}}\right) = \prod_{i=1}^{k}\frac{1-c-c\cdot\sum_{j\leqslant i}x_{rj}}{1-c-c\sum_{j<i}x_{rj}} = \frac{1-c-c\sum_{j\leqslant k}x_{rj}}{1-c} \geqslant \frac{1-2c}{1-c},$$

where second equality follows by the definition of α_{ri}, first inequality follows by Equation (15), and the last inequality by the fact that $\sum_{j\leqslant k} x_{rj} \leqslant 1$. ◻

We apply Lemma 3 to further simplify the lower bound of Equation (14) on $\mathbf{Pr}[u,v \text{ have witnesses}]$.

$$\mathbf{Pr}\left[u,v \text{ have witnesses}\right] \geqslant \sum_{\substack{u^*\neq v^*\\ u^*,v^*\notin\{u,v\}}} c^2 \cdot x_{uu^*}x_{vv^*} \cdot \left(\frac{1-2c}{1-c}\right)^2$$
$$= c^2 \cdot \left(\frac{1-2c}{1-c}\right)^2 \cdot \left(\sum_{s\notin\{u,v\}} x_{us} \cdot \sum_{s\notin\{u,v\}} x_{vs} - \sum_{s\notin\{u,v\}} x_{us}\cdot x_{vs}\right).$$

We recall that $\mathbf{Pr}[u,v \text{ are matched at } (uv)] \geqslant \mathbf{Pr}[u,v \text{ have witnesses}]$. Therefore, we get the following bound from Equation (12).

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\tilde{E}}[u,v \text{ are unmatched at } (uv)] &\geqslant 1-c\left(\sum_{s\notin\{u,v\}} x_{us} + \sum_{s\notin\{u,v\}} x_{vs}\right) + \mathop{\mathbf{Pr}}_{\tilde{E}}[u,v \text{ have witnesses}] \\ &\geqslant 1-c\left(\sum_s x_{us} + \sum_s x_{vs}\right) + c^2\cdot\left(\frac{1-2c}{1-c}\right)^2\cdot\left(\sum_s x_{us}\cdot\sum_s x_{vs} - \sum_s x_{us}\cdot x_{vs}\right), \quad (16)\end{aligned}$$

where all summations are taken over $s\in V\backslash\{u,v\}$. Note that for all s we have $x_{us}, x_{vs}\geqslant 0$, $x_{us}+x_{vs}\leqslant 1$, as well as $\sum_s x_{us}\leqslant 1$ and $\sum_s x_{vs}\leqslant 1$. Let us find the minimum of the function $f(\mathbf{x}) \stackrel{\text{def}}{=\!=}$ RHS of Equation (16). To conclude the proof it suffices to show that $\min_{\mathbf{x}} f(\mathbf{x})$ is at least the value in the statement of Lemma 1 (where the minimum is over all positive vectors satisfying Equation (7)).

Lemma 4.

$$f(\mathbf{x}) \geqslant 1-2c+\frac{c^2}{2}\cdot\left(\frac{1-2c}{1-c}\right)^2.$$

Proof. We observe that $\frac{\partial f}{\partial x_{ui}} = -c + c^2\cdot\left(\frac{1-2c}{1-c}\right)^2\cdot\left(\sum_s x_{vs} - x_{vi}\right) < 0$ for all $i\in V\backslash\{u,v\}$. Similarly, $\frac{\partial f}{\partial x_{vi}} < 0$ for all $i\in V\backslash\{u,v\}$. That means that the minimum of f is achieved at a boundary point $\mathbf{x}$, which does not allow us to increase any of the x_{vi} or x_{ui}. The analysis proceed in two cases.

Case (a). If $\sum_s x_{us} = \sum_s x_{vs} = 1$, then we can find a good upper bound on $\sum_s x_{us}\cdot x_{vs}$ as follows. First, $\frac{1}{4}\sum_s x_{us}\cdot x_{vs} \leqslant \frac{1}{4}\sum_s x_{us}\cdot(1-x_{us}) = \frac{1}{4}-\frac{1}{4}\sum_s x_{us}^2$. Similarly, $\frac{1}{4}\sum_s x_{us}\cdot x_{vs}\leqslant \frac{1}{4}-\frac{1}{4}\sum_s x_{vs}^2$. Second, $\frac{1}{2}\sum_s x_{us}\cdot x_{vs} \leqslant \frac{1}{4}\sum_s x_{us}^2 + \frac{1}{4}\sum_s x_{vs}^2$. Now, if we add the last three inequalities together, we get $\sum_s x_{us}\cdot x_{vs}\leqslant \frac{1}{2}$. Thus $f(\mathbf{x})\geqslant 1-2c+c^2\cdot\left(\frac{1-2c}{1-c}\right)^2\cdot(1-\frac{1}{2})$, which is equal to the desired bound in Lemma 1.

Case (b). If $\sum_s x_{us} < 1$, or $\sum_s x_{vs} < 1$. Then each inequality $x_{us}+x_{vs}\leqslant 1$ must be tight for every $s\in V\backslash\{u,v\}$. It means that $\sum_s x_{vs}+\sum_s x_{us} = \sum_s(x_{us}+x_{vs}) = \sum_s 1\in\mathbb{Z}$, also $\sum_s x_{vs}+\sum_s x_{us} < 2$. Therefore, $\sum_s x_{vs}+\sum_s x_{us}\leqslant 1$. We get that $f(\mathbf{x})\geqslant 1-c(\sum_s x_{us}+\sum_s x_{vs}) = 1-c \geqslant 1-2c+\frac{c^2}{2}\cdot\left(\frac{1-2c}{1-c}\right)^2$. ◻

5. Discussion In this paper we introduce a framework of batched prophet inequalities and apply it to stochastic online matching problems. Our results demonstrate the merit of online contention resolution schemes as a useful tool for generating prophet inequalities with good performance. The new framework suggests many fascinating avenues for future work. Some of them are listed below.

1. It would be interesting to study whether our algorithms apply to online matching problems with other arrival models. For example, upon the arrival of a vertex, all edges from the new vertex to all *future* vertices are revealed.
2. We achieve an optimal $\frac{1}{2}$-competitive (resp., 0.337-competitive) algorithm for vertex (resp., edge) arrival via OCRS. Are there pricing-based algorithms with comparable performance?
3. For both vertex and edge arrival settings, consider the random arrival order, a.k.a. prophet secretary. For the one-sided vertex arrivals, Ehsani et al. [19] showed that the competitive ratio can be improved to $1-\frac{1}{e}$. Does it generalize to two-sided vertex arrival model?
4. In the edge arrival setting, it seems unlikely that an OCRS can be better than 0.382-selectable due to the discussion in Section 4.2. Yet, the best upper bounds known for prophet inequality are $\frac{3}{7}$ and 0.420 against optimal fractional matching, and ex-ante relaxation, respectively. The gap is fairly large and it is unclear if OCRS approach can yield tight competitive ratio.
5. This paper focuses on matching feasibility constraints. Consider studying other natural batched prophet inequality settings, with natural structures of ordered partitions into batches.

References

[1] Adamczyk M, Wlodarczyk M (2018) Random order contention resolution schemes. *FOCS*, 790–801 (IEEE Computer Society).

[2] Alaei S (2011) Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *FOCS*, 512–521 (IEEE Computer Society).

[3] Alaei S, Hajiaghayi M, Liaghat V (2012) Online prophet-inequality matching with applications to ad allocation. *EC*, 18–35 (ACM).

[4] Alon N, Pollner T, Weinberg SM (2020) Three results on prophet inequalities on (hyper-) graphs. Personal communication.

[5] Ashlagi I, Burq M, Dutta C, Jaillet P, Saberi A, Sholley C (2019) Edge weighted online windowed matching. *EC*, 729–742 (ACM).

[6] Azar PD, Kleinberg R, Weinberg SM (2014) Prophet inequalities with limited information. *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, 1358–1377 (SIAM).

[7] Bruggmann S, Zenklusen R (2019) An optimal monotone contention resolution scheme for bipartite matchings via a polyhedral viewpoint. *CoRR* abs/1905.08658.

[8] Buchbinder N, Jain K, Naor J (2007) Online primal-dual algorithms for maximizing ad-auctions revenue. *ESA*, volume 4698 of *Lecture Notes in Computer Science*, 253–264 (Springer).

[9] Buchbinder N, Segev D, Tkach Y (2017) Online algorithms for maximum cardinality matching with edge arrivals. *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, 22:1–22:14, URL http://dx.doi.org/10.4230/LIPIcs.ESA.2017.22.

[10] Chawla S, Hartline JD, Malec DL, Sivan B (2010) Multi-parameter mechanism design and sequential posted pricing. *STOC*, 311–320 (ACM).

[11] Chekuri C, Vondrák J, Zenklusen R (2014) Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM J. Comput.* 43(6):1831–1879.

[12] Correa J, Dütting P, Fischer F, Schewior K (2019) Prophet inequalities for iid random variables from an unknown distribution. *Proceedings of the 2019 ACM Conference on Economics and Computation, EC*, 3–17 (ACM).

[13] Correa JR, Cristi A, Epstein B, Soto JA (2020) The two-sided game of googol and sample-based prophet inequalities. *SODA*, 2066–2081 (SIAM).

[14] Costello KP, Tetali P, Tripathi P (2012) Stochastic matching with commitment. *Automata, Languages, and Programming - International Colloquium (ICALP), Warwick, UK, July 9-13, 2012, Part I*, 822–833.

[15] Devanur NR, Jain K (2012) Online matching with concave returns. *STOC*, 137–144 (ACM).

[16] Duetting P, Feldman M, Kesselheim T, Lucier B (2017) Prophet inequalities made easy: Stochastic optimization by pricing non-stochastic inputs. *FOCS*, 540–551 (IEEE Computer Society).

[17] Dütting P, Kleinberg R (2015) Polymatroid prophet inequalities. *Algorithms-ESA 2015*, 437–449 (Springer).

[18] Eden A, Feldman M, Fiat A, Segal K (2018) An economic-based analysis of RANKING for online bipartite matching. *CoRR* abs/1804.06637.

[19] Ehsani S, Hajiaghayi M, Kesselheim T, Singla S (2018) Prophet secretary for combinatorial auctions and matroids. *SODA*, 700–714 (SIAM).

[20] Ezra T, Feldman M, Nehama I (2018) Prophets and secretaries with overbooking. *Proceedings of the 2018 ACM Conference on Economics and Computation, EC*, 319–320 (ACM).

[21] Feldman M, Gravin N, Lucier B (2015) Combinatorial auctions via posted prices. *SODA*, 123–135 (SIAM).

[22] Feldman M, Naor J, Schwartz R (2011) A unified continuous greedy algorithm for submodular maximization. *FOCS*, 570–579 (IEEE Computer Society).

[23] Feldman M, Svensson O, Zenklusen R (2016) Online contention resolution schemes. *SODA*, 1014–1033 (SIAM).

[24] Gamlath B, Kale S, Svensson O (2019) Beating greedy for stochastic bipartite matching. *SODA*, 2841–2854 (SIAM).

[25] Gamlath B, Kapralov M, Maggiori A, Svensson O, Wajc D (2019) Online matching with general arrivals. *FOCS*, 26–37 (IEEE Computer Society).

[26] Gravin N, Tang ZG, Wang K (2019) Online stochastic matching with edge arrivals. *CoRR* abs/1911.04686.

[27] Gravin N, Wang H (2019) Prophet inequality for bipartite matching: Merits of being simple and non adaptive. *Proceedings of the 2019 ACM Conference on Economics and Computation, EC*, 93–109.

[28] Guruganesh G, Lee E (2017) Understanding the correlation gap for matchings. *FSTTCS*, volume 93 of *LIPIcs*, 32:1–32:15 (Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik).

[29] Hajiaghayi MT, Kleinberg RD, Sandholm T (2007) Automated online mechanism design and prophet inequalities. *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, 58–65.

[30] Huang Z, Kang N, Tang ZG, Wu X, Zhang Y, Zhu X (2020) Fully online matching. *J. ACM* 67(3):17:1–17:25.

[31] Huang Z, Peng B, Tang ZG, Tao R, Wu X, Zhang Y (2019) Tight competitive ratios of classic matching algorithms in the fully online model. *SODA*, 2875–2886 (SIAM).

[32] Huang Z, Tang ZG, Wu X, Zhang Y (2020) Fully online matching ii: Beating ranking and water-filling. *FOCS* (IEEE Computer Society).

[33] Karp RM, Vazirani UV, Vazirani VV (1990) An optimal algorithm for on-line bipartite matching. *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, STOC*, 352–358 (ACM).

[34] Kennedy DP (1985) Optimal stopping of independent random variables and maximizing prophets. *The Annals of Probability* 566–571.

[35] Kennedy DP (1987) Prophet-type inequalities for multi-choice optimal stopping. *Stochastic Processes and their applications* 24(1):77–88.

[36] Kertz RP (1986) Comparison of optimal value and constrained maxima expectations for independent random variables. *Advances in applied probability* 18(2):311–340.

[37] Kleinberg R, Weinberg SM (2019) Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games and Economic Behavior* 113:97–115.

[38] Krengel U, Sucheston L (1977) Semiamarts and finite values. *Bull. Amer. Math. Soc.* 83(4):745–747.

[39] Krengel U, Sucheston L (1978) On semiamarts, amarts, and processes with finite value. *Advances in Prob* 4(197-266):1–5.

[40] Lee E, Singla S (2017) Maximum matching in the online batch-arrival model. *IPCO*, volume 10328 of *Lecture Notes in Computer Science*, 355–367 (Springer).

[41] Lee E, Singla S (2018) Optimal online contention resolution schemes via ex-ante prophet inequalities. *ESA*, volume 112 of *LIPIcs*, 57:1–57:14 (Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik).

[42] Lucier B (2017) An economic view of prophet inequalities. *SIGecom Exchanges* 16(1):24–47.

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

[44] Mehta A, Saberi A, Vazirani UV, Vazirani VV (2007) Adwords and generalized online matching. *J. ACM* 54(5):22.

[45] Rubinstein A (2016) Beyond matroids: Secretary problem and prophet inequality with general constraints. *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, 324–332.

[46] Rubinstein A, Wang JZ, Weinberg SM (2019) Optimal single-choice prophet inequalities from samples. *arXiv preprint arXiv:1911.07945* .

[47] Samuel-Cahn E, et al. (1984) Comparison of threshold stop rules and maximum for independent non-negative random variables. *the Annals of Probability* 12(4):1213–1216.

[48] Wang Y, Wong SC (2015) Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. *ICALP (1)*, volume 9134 of *Lecture Notes in Computer Science*, 1070–1081 (Springer).

Appendix A: Extension to Fractional OCRS and Prophet Inequality In this section we extend the definition of batched OCRS to fractional-batched OCRS, and extend the reduction in Theorem 1 between batched prophet inequality against the fractional optimum and fractional-batched OCRS.

A.1. Fractional Batched OCRS Consider a *fractional sampling scheme* R that selects a random fractional sets, i.e., vectors $\mathbf{r} \in [0,1]^E$, where $\mathbf{r} = (\mathbf{r}^t)_{t\in[T]}$ is composed of mutually independent samples $\mathbf{r}^t \in [0,1]^{B_t}$ (recall $E = \bigsqcup_t B_t$).

Definition 5 (c-fractional-selectable batched OCRS). An online selection algorithm ALG with respect to a fractional sampling scheme R is a batched OCRS if it selects a set $I_t \subseteq B_t$ at every time t such that $I \overset{\text{def}}{=\!=} \bigsqcup_{t\in[T]} I_t \in \mathcal{M}$ is feasible. It is a c-batched-OCRS if:

$$\Pr_{\mathsf{ALG}}\left[e \in I_t \mid \mathbf{r}^t = \mathbf{s}\right] \geqslant c \cdot \mathbf{s}_e \quad \text{ for all } t \in [T], \mathbf{s} \in [0,1]^{B_t}, \text{ and } e \in B_t. \tag{17}$$

The algorithm ALG is oblivious to the partition into batches and to the arrival order of the batches. It only knows the general structure $\mathcal{B}$. Thus, at time t, ALG chooses I_t based on $B_1, \ldots, B_t$, and $\mathbf{r}^1, \ldots, \mathbf{r}^t$.

A.2. Batched Fractional Prophet Inequality For certain feasibility constraints $\mathcal{M}$, it makes sense to consider fractional optimum $\mathsf{f\text{-}OPT}(\mathbf{w}) \in [0,1]^E$, where $\mathsf{f\text{-}OPT} \in \mathcal{F}_{\mathcal{M}}$ for a fractional relaxation of feasibility family $\mathcal{M}$ and $\mathsf{f\text{-}OPT} = \arg\max_{\mathbf{x}\in\mathcal{F}_{\mathcal{M}}} \langle \mathbf{w}, \mathbf{x}\rangle$. The weight of $\mathsf{f\text{-}OPT}$ is $\mathbf{w}(\mathsf{f\text{-}OPT}(\mathbf{w})) \overset{\text{def}}{=\!=} \langle \mathbf{w}, \mathsf{f\text{-}OPT}(\mathbf{w})\rangle \geqslant \mathbf{w}(\mathsf{OPT}(\mathbf{w}))$.

Definition 6 (c-batched-fractional-prophet inequality). A batched-prophet inequality algorithm ALG is an online selection process that selects at time t a set $I_t \subseteq B_t$ such that $I \overset{\text{def}}{=\!=} \bigsqcup_{t\in[T]} I_t$ is feasible (i.e., $I \in \mathcal{M}$). We say that ALG has competitive ratio c against fractional optimum if

$$\mathop{\mathbf{E}}_{\mathbf{w},I}\left[\mathbf{w}(I)\right] \geqslant c \cdot \mathop{\mathbf{E}}_{\mathbf{w}}\left[\mathbf{w}(\mathsf{f\text{-}OPT}(\mathbf{w}))\right]. \tag{18}$$

A.3. Reduction: Prophet Inequality to OCRS We define the fractional random sampling scheme $R(\mathbf{w}, \mathbf{F})$ for $\mathbf{w} \sim \mathbf{F}$ as follows. Let $\mathbf{r}^t(\mathbf{w}^t, \mathbf{F}) \overset{\text{def}}{=\!=} \mathsf{f\text{-}OPT}(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)})\big|_{B_t}$, where $\widetilde{\mathbf{w}}^{(t)} \sim \mathbf{F}_{-t}$ is independently generated of $\mathbf{w}$, i.e., $\mathbf{r}^t(\mathbf{w}^t, \mathbf{F})_e = \mathsf{f\text{-}OPT}(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)})_e$ for all $e \in B_t$. Let $\mathbf{r}(\mathbf{w}, \mathbf{F}) \overset{\text{def}}{=\!=} (\mathbf{r}^t(\mathbf{w}^t, \mathbf{F}))_{t\in[T]}$. We notice that

1. The distribution of $\mathbf{r}(\mathbf{w}, \mathbf{F}) \sim R(\mathbf{w}, \mathbf{F})$ is a product distribution over the random variables $\mathbf{r}^t(\mathbf{w}^t, \mathbf{F})$.
2. Since $\mathbf{F}$ is a product distribution, $(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)}) \sim \mathbf{F}$.
3. $\forall t \in [T]$, $\mathbf{r}^t(\mathbf{w}^t, \mathbf{F})$ has the same distribution as $\mathsf{f\text{-}OPT}(\mathbf{w})\big|_{B_t}$ (i.e., restriction to $e \in B_t$), where $\mathbf{w} \sim \mathbf{F}$.
4. For every $t \in [T]$,

$$\mathop{\mathbf{E}}_{\mathbf{w},R}\left[\langle \mathbf{w}^t, \mathbf{r}^t(\mathbf{w}^t, \mathbf{F})\rangle\right] = \mathop{\mathbf{E}}_{\mathbf{w}}\left[\left\langle \mathbf{w}^t, \mathsf{f\text{-}OPT}(\mathbf{w})\big|_{B_t}\right\rangle\right]. \tag{19}$$

Theorem 5 **(reduction from prophet inequality to OCRS (fractional))**. *For every set $\mathcal{B}$ of feasible ordered partitions, given a c-batched OCRS for the fractional sampling scheme $R(\mathbf{w}, \mathbf{F})$ with $\mathbf{w} \sim \mathbf{F}$, one can construct a batched prophet inequality c-competitive algorithm for $\mathbf{w} \sim \mathbf{F}$ against the fractional optimum.*

Proof. Consider the following online algorithm:

Algorithm 4 Reduction from c-batched prophet inequality to fractional c-batched OCRS

1: **for** $t \in \{1, ..., T\}$ **do**
2: Let $\mathbf{w}^t$ be the weights of elements in B_t
3: Resample the weights $\widetilde{\mathbf{w}}^{(t)} \sim \mathbf{F}_{-t}$
4: Let $\mathbf{r}^t \leftarrow \mathsf{f\text{-}OPT}(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)})\big|_{B_t}$ (i.e., $\mathbf{r}^t_e = \mathsf{f\text{-}OPT}(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)})_e$ for each $e \in B_t$).
5: $I_t \leftarrow c\text{-OCRS}(B_1, \ldots, B_t, \mathbf{r}^1, \ldots, \mathbf{r}^t)$
6: **end for**
7: Return $I = \bigsqcup_{t \in [T]} I_t$

Without loss of generality, we may assume that the values of the sampling scheme are discretized, i.e., there are only countably many values in $[0,1]^{B_t}$ that $\mathbf{r}^t$ can take. Then

$$\begin{aligned}
\mathop{\mathbf{E}}_{\mathbf{w},R,I}[\mathbf{w}(I)] &= \sum_{t\in[T]} \mathop{\mathbf{E}}_{\mathbf{w},R,I}[\mathbf{w}(I_t)] \\
&= \sum_{t\in[T]} \sum_{\mathbf{s}\in[0,1]^{B_t}} \mathop{\mathbf{E}}_{\mathbf{w},R,I}\left[\mathbf{w}(I_t) \;\middle|\; \mathbf{r}^t = \mathbf{s}\right] \mathop{\mathbf{Pr}}_{\mathbf{w},R}\left[\mathbf{r}^t = \mathbf{s}\right] \\
&= \sum_{t\in[T]} \sum_{\mathbf{s}\in[0,1]^{B_t}} \mathop{\mathbf{E}}_{\mathbf{w},I}\left[\sum_{e\in B_t} w_e \cdot \mathop{\mathbf{Pr}}_{I}[e \in I_t] \;\middle|\; \mathbf{r}^t = \mathbf{s}\right] \mathop{\mathbf{Pr}}_{\mathbf{w}^t,\mathbf{r}^t}\left[\mathbf{r}^t = \mathbf{s}\right] \\
&\overset{(17)}{\geqslant} \sum_{t\in[T]} \sum_{\mathbf{s}\in[0,1]^{B_t}} \mathop{\mathbf{E}}_{\mathbf{w}^t}\left[\sum_{e\in B_t} w_e \cdot c \cdot \mathbf{s}_e \;\middle|\; \mathbf{r}^t = \mathbf{s}\right] \mathop{\mathbf{Pr}}_{\mathbf{w}^t,\mathbf{r}^t}\left[\mathbf{r}^t = \mathbf{s}\right] \\
&= c \sum_{t\in[T]} \sum_{\mathbf{s}\in[0,1]^{B_t}} \mathop{\mathbf{E}}_{\mathbf{w}^t}\left[\langle \mathbf{w}^t, \mathbf{s}\rangle \mid \mathbf{r}^t = \mathbf{s}\right] \mathop{\mathbf{Pr}}_{\mathbf{w}^t,\mathbf{r}^t}\left[\mathbf{r}^t = \mathbf{s}\right] \\
&= c \sum_{t\in[T]} \mathop{\mathbf{E}}_{\mathbf{w}^t,\mathbf{r}^t}\left[\langle \mathbf{w}^t, \mathbf{r}^t\rangle\right] \\
&\overset{(19)}{=} c \sum_{t\in[T]} \mathop{\mathbf{E}}_{\mathbf{w}}\left[\left\langle \mathbf{w}^t, \mathsf{f\text{-}OPT}(\mathbf{w})\big|_{B_t}\right\rangle\right] = c \cdot \mathop{\mathbf{E}}_{\mathbf{w}}\left[\langle \mathbf{w}, \mathsf{f\text{-}OPT}(\mathbf{w})\rangle\right],
\end{aligned}$$

where the third equality holds since I_t and $\mathbf{w}$ are independent given that $\mathbf{r}^t = \mathbf{s}$. □

Appendix B: Stronger Benchmarks for Batched Prophet Inequality for Matching
In this section we show that our results for both vertex and edge arrival models hold against stronger benchmarks than the optimal integral matching. Specifically, for the vertex arrival model we establish guarantees against the optimal fractional matching, and for the edge arrival model, we establish guarantees against the even stronger benchmark of optimal ex-ante matching. The set of *fractional matchings* $\mathbf{y} = (y_e)_{e\in E}$ can be specified by the matching polytope

$$\mathcal{F}_{\mathcal{M}} \overset{\text{def}}{=\!=} \{\mathbf{y} \mid \forall v \in V \quad \sum_{u\in V} y_{(uv)} \leqslant 1, \;\; \forall e \in E \quad y_e \geqslant 0\}.$$

B.1. Vertex Arrival: Fractional Optimum Let $\mathsf{f\text{-}OPT}(\mathbf{w}) \overset{\text{def}}{=\!=} \arg\max_{\mathbf{y}\in\mathcal{F}_{\mathcal{M}}} \langle \mathbf{w}, \mathbf{y}\rangle$ be the optimal fractional matching. Note that

$$\mathbf{w}(\mathsf{f\text{-}OPT}(\mathbf{w})) \overset{\text{def}}{=\!=} \langle \mathbf{w}, \mathsf{f\text{-}OPT}(\mathbf{w})\rangle \geqslant \mathbf{w}(\mathsf{OPT}(\mathbf{w})).$$

In this section we show that our result for the vertex arrival model holds against the stronger benchmark of $\mathsf{f\text{-}OPT}(\mathbf{w})$, namely we can strengthen the guarantee in Definition 2 to

$$\mathop{\mathbf{E}}_{\mathbf{w},I}[\mathbf{w}(I)] \geqslant c \cdot \mathop{\mathbf{E}}_{\mathbf{w}}[\mathbf{w}(\mathsf{f\text{-}OPT}(\mathbf{w}))].$$

Let $(B_t)_{t\in[T]}$ be a feasible ordered partition in $\mathcal{B}^{v.a.}$. It induces a fractional random sampling scheme R with respect to the fractional optimum $\mathsf{f\text{-}OPT}$ that generates vector $\mathbf{r}(\mathbf{w},\mathbf{F})\in[0,1]^E$ as defined in Section A.3. Let $x_{uv}^{\mathsf{f\text{-}OPT}}=\mathbf{E}[\mathbf{r}_{(uv)}]$, and let $\mathbf{x}^{\mathsf{f\text{-}OPT}}=(x_{uv}^{\mathsf{f\text{-}OPT}})_{(uv)\in E}$.

Observe that for any edge (uv), $x_{uv}^{\mathsf{f\text{-}OPT}}=\mathbf{E}[\mathsf{f\text{-}OPT}(\mathbf{w})_{(uv)}]$, where $\mathbf{w}\sim\mathbf{F}$. Therefore, $\mathbf{x}^{\mathsf{f\text{-}OPT}}\in\mathcal{F}_{\mathcal{M}}$ (recall that $\mathcal{F}_{\mathcal{M}}=\{\mathbf{y}\mid\forall v\in V\quad\sum_{u\in V}y_{(uv)}\leqslant 1,\ \forall e\in E\quad y_e\geqslant 0\}$). We also observe that

$$\sum_{u<v}\mathbf{r}^v_{(uv)}\leqslant 1\quad\text{for every } v\in V \text{ and every realization } \mathbf{r}^v\in[0,1]^{B_v},\tag{20}$$

since $\mathbf{r}^v$ is a projection of a fractional matching on B_v.

With these two properties, in Appendix C we construct a $\frac{1}{2}$-batched fractional OCRS for vertex arrival model, which implies a $\frac{1}{2}$-batched fractional prophet inequality for the maximum fractional matching.

B.2. Edge Arrival: Ex-ante Optimum As was previously observed by [41], for the special case in which each batch consists of a single element, one can provide the guarantees with respect to the stronger benchmark of the optimal ex-ante solution. The optimal ex-ante solution $\mathbf{y}$ is defined as follows:

$$\mathbf{y}=\arg\max\sum_e\mathop{\mathbf{E}}_{w_e}\left[w_e|w_e\geqslant F_e^{-1}(1-y_e)\right]\cdot y_e\qquad\text{subject to } \mathbf{y}\in\mathcal{F}_{\mathcal{M}}.$$

Let $R^{ex-ante}(\mathbf{w},\mathbf{F})=\{e\mid w_e\geqslant F_e^{-1}(1-y_e)\}$.

By definition, the distribution of $R^{ex-ante}(\mathbf{w},\mathbf{F})$ is a product distribution of $R_e^{ex-ante}(w_e,F_e)$. Therefore, any c-OCRS with respect to $R^{ex-ante}(\mathbf{w},\mathbf{F})$ gives us a prophet inequality algorithm with competitive ratio c with respect to the optimal ex-ante solution. Specifically, our 0.337-OCRS from Section 4 when applied to $R^{ex-ante}$ implies a 0.337-competitive algorithm for the prophet inequality problem against the ex-ante optimum. Unfortunately, the reduction from general *batched* prophet inequalities to *batched* OCRSs does not work for ex-ante benchmark. E.g., it does not even work for the vertex arrival setting of our paper.

Appendix C: A 1/2-Batched OCRS for Fractional Matching with Vertex Arrival In what follows we extend our construction from Section 3 to a 1/2-batched fractional OCRS for vertex arrival. Let $\mathbf{r}=(\mathbf{r}^1,\dots\mathbf{r}^{|V|})$ be independent random variables over $[0,1]^{B_i}$. Let $x_{uv}=\mathbf{E}[\mathbf{r}^v_{(uv)}]$, and $\mathbf{x}=(x_{uv})_{(uv)\in E}$.

We write $u<v$ if vertex u arrive before vertex v.

THEOREM 6. *If $\mathbf{r}$ satisfies the following two conditions:*

$$\sum_u x_{uv}\leqslant 1\quad\textit{for every } v\in V\tag{21}$$

$$\sum_{u<v}\mathbf{r}^v_{(uv)}\leqslant 1\quad\textit{for every } v\in V \textit{ and every realization } \mathbf{r}^v\in[0,1]^{B_v}\tag{22}$$

Then, $\mathbf{r}$ admits a 1/2-batched fractional OCRS for the $\mathcal{B}^{v.a.}$ structure of batches..

Note that $\mathbf{r}$ as defined in Appendix B for the vertex arrival model satisfies Equations (21),(22).

Proof. Upon the arrival of a vertex v, we compute $\alpha_u(v)$ for every $u<v$ as follows:

$$\alpha_u(v)\stackrel{\text{def}}{=\!=}\frac{1}{2-\sum_{z<v}x_{uz}}\leqslant\frac{1}{2-\sum_z x_{uz}}\stackrel{(21)}{\leqslant}1.\tag{23}$$

Note that $\alpha_u(v)$ cannot be calculated before the arrival of v. We claim that the following algorithm is a $\frac{1}{2}$-batched fractional OCRS with respect to $\mathbf{r}$:

Algorithm 5 1/2-batched fractional OCRS for vertex arrival

for $v \in \{1, ..., |V|\}$ **do**
  Calculate $x_{uz} = \mathbf{Pr}[(uv) \in R]$ for all $u, z < v$ and $\alpha_u(v)$ for all $u < v$.
  Among all unmatched $u < v$, choose one u (or none) with probability $\mathbf{r}^v_{(uv)} \cdot \alpha_u(v)$.
  If u was chosen, then match (uv).
end for

We first show that Algorithm 5 is well defined; namely, that (i) Algorithm 5 matches not more than one edge incident to u and v, and (ii) for every v, we can match each available vertex $u < v$ with probability $\mathbf{r}^v_{(uv)} \cdot \alpha_u(v)$. The worst case is where all previous vertices are available. Thus, a sufficient condition is that $\sum_{u<v} \mathbf{r}^v_{(uv)} \cdot \alpha_u(v) \leqslant 1$. Indeed,

$$\sum_{u<v} \mathbf{r}^v_{(uv)} \cdot \alpha_u(v) \overset{(23)}{\leqslant} \sum_{u<v} \mathbf{r}^v_{(uv)} \overset{(22)}{\leqslant} 1.$$

It remains to show that Algorithm 5 is a 1/2-batched fractional OCRS with respect to $\mathbf{r}$. We prove that $\mathbf{Pr}[(uv) \text{ is matched}] = \frac{x_{uv}}{2}$ by induction on the set of vertices V, according to their arrival order. The base of the induction ($V = \emptyset$) holds trivially. For the induction step, assume that $\mathbf{Pr}[(uz) \text{ is matched}] = \frac{x_{uz}}{2}$ for all $u, z < v$. We show that $\mathbf{Pr}[(uv) \text{ is matched}] = \frac{x_{uv}}{2}$ for all $u < v$. In what follows, we say that "u is unmatched *at* v" if u is unmatched *right before* v arrives.

$$\mathbf{Pr}\left[u \text{ is unmatched at } v\right] = 1 - \sum_{z<v} \mathbf{Pr}\left[(uz) \text{ is matched}\right] = 1 - \frac{1}{2}\sum_{z<v} x_{uz}. \tag{24}$$

Therefore,

$$\begin{aligned} \mathbf{Pr}\left[(uv) \text{ is matched}\right] &= \mathbf{Pr}\left[u \text{ is unmatched at } v\right] \cdot \mathbf{E}\left[\mathbf{r}^v_{(uv)} \cdot \alpha_u(v)\right] \\ &\overset{(23),(24)}{=} \left(1 - \frac{1}{2}\sum_{z<v} x_{uz}\right) \cdot \frac{1}{2 - \sum_{z<v} x_{uz}} \cdot x_{uv} \\ &= \frac{x_{uv}}{2}. \end{aligned}$$

To conclude the proof that Algorithm 5 is a $\frac{1}{2}$-batched fractional OCRS with respect to $\mathbf{r}$, we show that for every $u < v$ and every $[0,1]^{B_v}$

$$\mathbf{Pr}\left[(uv) \in I_v \mid \mathbf{r}^v = \mathbf{s}\right] = \mathbf{Pr}\left[u \text{ is unmatched at } v\right] \cdot \mathbf{s}_{(uv)} \cdot \alpha_u(v)$$
$$\overset{(23),(24)}{=} \left(1 - \frac{1}{2}\sum_{z<v} x_{uz}\right) \cdot \frac{1}{2 - \sum_{z<v} x_{uz}} \cdot \mathbf{s}_{(uv)} = \frac{\mathbf{s}_{(uv)}}{2}. \qquad \square$$

Appendix D: Upper Bounds for Matching Prophet Inequality In this section we present upper bounds on the competitive ratios for matching prophet inequality with edge arrival, with respect to the fractional and ex-ante optimal solutions. Note that the 1/2 competitive ratio with respect to the classical prophet inequality extends trivially to matching prophet inequality (for both vertex and edge arrival models), implying that our 1/2 competitive ratio for vertex arrival, as implied from Section 3, is tight. The following propositions give upper bounds on the competitive ratio of prophet inequalities for matching with edge arrival. Proposition 1 gives an upper bound with respect to the optimal fractional matching, and Proposition 2 gives an upper bound with respect to the optimal ex-ante matching (see definition below).

Proposition 1. *Under the edge arrival model, no online algorithm can get better than $\frac{3}{7}$ of f-OPT, even for 6-vertex graphs.*

Proof. Consider the graph depicted in Figure 1(a) with 6 vertices a, b, c, d, e, f, where edges (ab), (bc), (ac), and $(de), (ef), (df)$ have a fixed weight of 1, and all other 9 edges have weight $\frac{1}{4\epsilon}$ with probability ϵ (for an arbitrarily small ϵ), and 0 otherwise. We refer to the latter edges as the big edges. Suppose the 6 fixed edges arrive first, followed by the big edges.

The optimal fractional solution is the following: if there exists a big edge (this happens with probability $9\epsilon + O(\epsilon^2)$), then take it; else take each of the fixed edges with probability 1/2. This approximately gives us $9\epsilon\frac{1}{4\epsilon} + (1-9\epsilon)3 = \frac{21}{4}$.

We next show that any online algorithm gets at most $\frac{9}{4}$, resulting in a ratio of $\frac{3}{7}$, as claimed. An online algorithm can choose to select either 0, 1, or 2 fixed edges, without knowing the realization of the big edges. If it chooses 0 fixed edges, it gets $\sim 9\epsilon\frac{1}{4\epsilon} = \frac{9}{4}$. If it chooses 1 fixed edge, it gets $\sim 1 + 3\epsilon\frac{1}{4\epsilon} = \frac{7}{4}$. If it chooses 2 fixed edges (one from each triangle), it gets $\sim 2 + \epsilon\frac{1}{4\epsilon} = \frac{9}{4}$. This completes the proof. $\square$

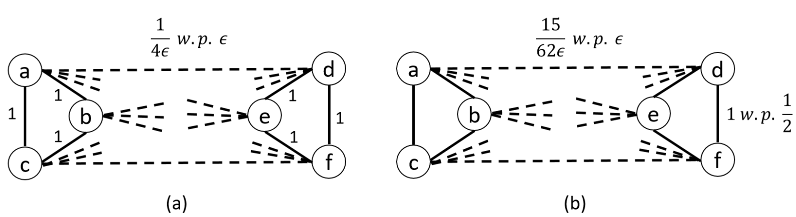

Figure 1. Upper bounds for matching prophet inequality with edge arrival. (a) upper bound w.r.t. optimal fractional matching. Solid lines have weight 1; dotted lines have weight $1/4\epsilon$ w.p. ϵ. (b) upper bound w.r.t. optimal ex-ante matching. Solid lines have weight 1 w.p. 1/2; dotted lines have weight $15/62\epsilon$ w.p. ϵ.

A stronger benchmark than the optimal fractional matching is the optimal ex-ante matching $\mathbf{y} \in [0,1]^E$, defined as follows:

$$\mathbf{y} = \arg\max \sum_e \mathbf{E}_{w_e}\left[w_e \mid w_e \geqslant F_e^{-1}(1-y_e)\right] \cdot y_e \qquad \text{subject to } \mathbf{y} \in \mathcal{F}_{\mathcal{M}}$$
$$\textsf{ex-ante-OPT}(\mathbf{F}) = \sum_e \mathbf{E}_{w_e}\left[w_e \mid w_e \geqslant F_e^{-1}(1-y_e)\right].$$

The following proposition gives an upper bound with respect to the optimal ex-ante matching. Note that our lower bounds for edge arrival apply also with respect to the optimal ex-ante solution (see Section B.2).

PROPOSITION 2. *Under the edge arrival model, no online algorithm can get better than $\frac{135}{321}$ of ex-ante-OPT, even for 6-vertex graphs.*

Proof. Consider the graph depicted in Figure 1(b) with 6 vertices a, b, c, d, e, f, where edges (ab), (bc), (ac), and (de), (ef), (df) have a weight of 1 with probability $\frac{1}{2}$ and 0 otherwise. All other 9 edges have weight $\frac{15}{62\epsilon}$ with probability ϵ (for an arbitrarily small ϵ), and 0 otherwise. We refer to the latter edges as the big edges. Suppose the edges (ab), (bc), (ac) arrive first, followed by the edges (de), (ef), (df), and only then the big edges arrive.

The optimal *ex-ante* solution is the following: it takes the big edges with probability ϵ and the other edges with probability of approximately $\frac{1}{2}$. This gives approximately a value of $9\epsilon\frac{15}{62\epsilon} + 3 = \frac{321}{62}$.

We next show that any online algorithm gets at most $\frac{135}{62}$, resulting in a ratio of $\frac{135}{321}$, as claimed. An online algorithm can choose to select either 0 or 1 edges from the set $\{(ab),(bc),(ca)\}$ without knowing the realization of the big edges. If it chooses none of the edges $(ab),(bc),(ca)$, it gets $\frac{1}{8}\cdot 9\epsilon\cdot\frac{15}{62\epsilon} + \frac{7}{8}\max(9\epsilon\cdot\frac{15}{62\epsilon}, 1+3\epsilon\cdot\frac{15}{62\epsilon}) + O(\epsilon) = \frac{135}{62} + O(\epsilon)$. If it chooses one edge from $\{(ab),(bc),(ca)\}$, it gets $\sim 1 + \frac{1}{8}\cdot 3\epsilon\cdot\frac{15}{62\epsilon} + \frac{7}{8}\max(3\epsilon\frac{15}{62\epsilon}, 1+\epsilon\frac{15}{62\epsilon}) + O(\epsilon) = \frac{135}{62} + O(\epsilon)$. This completes the proof. □

In Appendix G we establish an improved upper bound for the setting of multigraphs.

Appendix E: Other definition of batched-OCRS: bad example Here, we discuss why a natural generalization of the previous OCRS for singletons to batched-OCRS in which one simply requires that $\mathbf{Pr}_{R,I}[e \in I] \geqslant c \cdot x_e$ instead of Equation (1) might be problematic. In particular, the standard reduction from a c-selectable OCRS to c-competitive prophet inequality ([23]) might not work in the batched setting for such definition of c-selectable batched-OCRS.

The problem with the standard reduction in batched settings is that there might be a dependency between the realized set and its weight. Our more complex approach (Equation (1)) requires the appropriate condition to hold for every realized set, thus allows us to account for the weight given any realized set.

To illustrate this point, consider the following example with 4 elements $\{1,2,3,4\}$ and the downward closed family of feasible sets $\{\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\},\{1\},\{2\},\{3\},\{4\},\varnothing\}$, i.e., the maximal feasible sets have all possible 2 element subsets of $\{1,2,3,4\}$ except the subset $\{1,2\}$ (which is a matroid). The elements arrive in two fixed batches: $B_1 = \{1,2\}$ and $B_2 = \{3,4\}$. We also consider a respective Prophet Inequality setting, in which all elements have weights independently distributed according to $\mathbf{w} \sim \mathbf{F} = \prod_{i=1}^{4} F_i$, where

$$F_1 = F_2 : \underset{w\sim F_1}{\mathbf{Pr}}[w = \varepsilon] = 1 \qquad \text{and} \qquad F_3 = F_4 : \underset{w\sim F_3}{\mathbf{Pr}}[w = 1] = 0.5, \quad \underset{w\sim F_3}{\mathbf{Pr}}[w = 0] = 0.5,$$

for some very small ε. The optimum solution $\mathsf{OPT}(\mathbf{w})$ picks the set $\{3,4\}$ if $w_3 = w_4 = 1$, and otherwise picks a set of size 2 with exactly one element among $\{1,2\}$ and the larger element among $\{3,4\}$. The expected weight of the optimum is

$$\underset{\mathbf{w}\sim\mathbf{F}}{\mathbf{E}}[\mathbf{w}(\mathsf{OPT}(\mathbf{w}))] = \underset{\mathbf{w}\sim\mathbf{F}}{\mathbf{E}}[w_3] + \underset{\mathbf{w}\sim\mathbf{F}}{\mathbf{E}}[w_4] + O(\varepsilon) = 1 + O(\varepsilon).$$

The standard sampling scheme $R = R_1 \sqcup R_2$ for the reduction from the Prophet inequality to OCRS observes the weights in the current batch and resample the weights of the remaining elements $\widetilde{\mathbf{w}}^{(t)} \sim \mathbf{F}_{-t}$; then it takes the set $R_t = B_t \cap \mathsf{OPT}(\mathbf{w}^t, \widetilde{\mathbf{w}}^{(t)})$ for $t \in \{1,2\}$. In our case,

$$R_1 = \begin{cases} \{1\} & \text{with probability } \frac{3}{8} \\ \{2\} & \text{with probability } \frac{3}{8} \\ \varnothing & \text{with probability } \frac{1}{4} \end{cases} \qquad R_2 = \begin{cases} \{3\} & \text{with probability } \frac{3}{8} \\ \{4\} & \text{with probability } \frac{3}{8} \\ \{3,4\} & \text{with probability } \frac{1}{4} \end{cases}$$

The marginal probability of the elements to be sampled in R are as follows:

$$x_1 = \mathbf{Pr}[1 \in R] = x_2 = \frac{3}{8} \qquad x_3 = \mathbf{Pr}[3 \in R] = x_4 = \frac{5}{8}.$$

For such a sampling scheme R one can achieve a pretty good c-selectable OCRS with $c=\frac{17}{20}$, by following a simple greedy algorithm that includes as many elements from R_t into a feasible set I as it can at each stage t. In particular, this greedy OCRS would always select elements 1 and 2, whenever 1, or 2 are included in R_1, i.e.,

$$\Pr_{R,I}[1\in I]=\Pr_{R,I}[1\in R]=x_1 \qquad \Pr_{R,I}[2\in I]=\Pr_{R,I}[2\in R]=x_2.$$

Sometimes greedy algorithm won't be able to take both 3 and 4 into I if $R_1\neq\varnothing$, in which case it will flip a coin and take one of the 3 or 4 uniformly at random. Thus to calculate $\mathbf{Pr}[3\in I]$ (similarly $\mathbf{Pr}[4\in I]$) we consider two cases $R_1=\varnothing$ and $|R_1|=1$ and get

$$\begin{aligned}\mathbf{Pr}[3\in I]=\mathbf{Pr}[R_1=\varnothing]\cdot\Big(&\mathbf{Pr}[R_2=\{3,4\}\mid R_1=\varnothing]+\mathbf{Pr}[R_2=\{3\}\mid R_1=\varnothing]\Big)\\ &+\mathbf{Pr}[|R_1|=1]\cdot\left(\frac{1}{2}\mathbf{Pr}[R_2=\{3,4\}\mid |R_1|=1]+\mathbf{Pr}[R_2=\{3\}\mid |R_1|=1]\right)\\ &=\frac{1}{4}\cdot\left(\frac{1}{4}+\frac{3}{8}\right)+\frac{3}{4}\cdot\left(\frac{1}{2}\cdot\frac{1}{4}+\frac{3}{8}\right)=\frac{17}{32}=\frac{17}{20}\cdot\frac{5}{8}=\frac{17}{20}\cdot x_3.\end{aligned}$$

Now, if we try to convert this greedy $\frac{17}{20}$-selectable OCRS into a prophet inequality algorithm ALG that selects a set I with the matching marginal probabilities $\mathbf{Pr}_I[1\in I]=\mathbf{Pr}_I[2\in I]=\frac{3}{8}$, then its competitive ratio will be noticeably smaller than $\frac{17}{20}$. Indeed,

$$\begin{aligned}\mathop{\mathbf{E}}_{\mathbf{w},I}[\mathsf{ALG}(\mathbf{w})]=\mathbf{Pr}[I\cap B_1=\varnothing]\cdot&\mathop{\mathbf{E}}_{\mathbf{w},I}[w_3+w_4\mid I\cap B_1=\varnothing]\\ &+\mathbf{Pr}[|I\cap B_1|=1]\cdot\mathop{\mathbf{E}}_{\mathbf{w},I}[\max(w_3,w_4)\mid |I\cap B_1|=1]+O(\varepsilon)\\ &=\frac{1}{4}\cdot\mathop{\mathbf{E}}_{\mathbf{w}\sim\mathbf{F}}[w_3+w_4]+\frac{3}{4}\cdot\mathop{\mathbf{E}}_{\mathbf{w}\sim\mathbf{F}}[\max(w_3,w_4)]+O(\varepsilon)\\ &=\frac{1}{4}\cdot(\mathbf{E}[w_3]+\mathbf{E}[w_4])+\frac{3}{4}\cdot 1\cdot\mathbf{Pr}[w_3=1\text{ or }w_4=1]+O(\varepsilon)\\ &=\frac{1}{4}\cdot 1+\frac{3}{4}\cdot\frac{3}{4}+O(\varepsilon)=\frac{13}{16}+O(\varepsilon)=\left(\frac{13}{16}+O(\varepsilon)\right)\cdot\mathop{\mathbf{E}}_{\mathbf{w}\sim\mathbf{F}}[\mathbf{w}(\mathsf{OPT}(\mathbf{w}))],\end{aligned}$$

i.e., the corresponding algorithm is only $\frac{13}{16}$-competitive, while we would like to have c-competitive algorithm with the same $c=\frac{17}{20}$ as the c-selectable OCRS we constructed before.

Appendix F: Pricing Approach: $\frac{1}{4}$ upper bound In this appendix we present a natural extension of the pricing-based algorithm of Feldman et al. [21] to the case of two-sided vertex arrival in bipartite matching, and show that it does not achieve a competitive ratio better than $\frac{1}{4}$. Upon arrival of a vertex v, the algorithm sets its price p_v to be a half of the expected future contribution (to the optimum matching) of future edges incident to v. It then considers an edge (uv) only if its weight covers the sum of the prices of its end points (i.e., $w_{uv}>p_u+p_v$). Among those, it chooses the one that maximizes $w_{uv}-p_u-p_v$. This algorithm appears as Algorithm 6 below, where $OPT(\mathbf{w})$ denotes the max-weight matching under weights $\mathbf{w}$, and $u<v$ denotes that vertex u arrives before vertex v.

Algorithm 6 Dynamic Pricing Algorithm

Let $p_v=\frac{1}{2}\sum_{u>v}\mathbf{E}_{\mathbf{w}\sim\mathbf{F}}[w_{uv}\cdot\mathbb{I}[(uv)\in\mathsf{OPT}(\mathbf{w})]]$.
Let $k\in\arg\max_{u<v,u\text{ unmatched}}\{w_{uv}-p_u\}$.
If $w_{vk}-p_u\geqslant p_v$, then include (vk) in the matching.

The example depicted in Figure 2 shows that the competitive ratio of Algorithm 6 is at most $\frac{1}{4}$. In this example, the expected maximum weight matching is $4-\frac{4\epsilon}{1+\epsilon}$ (by taking edge (cd) if

$w_{cd} > 0$, and taking (ac) otherwise). Suppose the arrival order is a, b, c, d. The prices calculated according to Algorithm 6 under this arrival order are $p_a = \frac{1-\epsilon}{1+\epsilon}, p_b = 0, p_c = 1, p_d = 0$. Given these prices, Algorithm 6 always chooses the edge (bc), which gives approximately $\frac{1}{4}$ of the expected maximum weight matching.

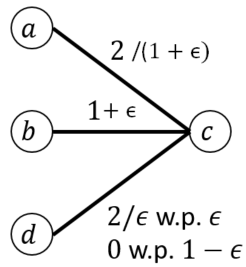

FIGURE 2. An upper bound of 1/4 on the pricing-based algorithm (Algorithm 6) for max-weight matching with two-sided vertex arrivals.

Appendix G: Extending to Multigraphs In this section we study multigraphs in the edge arrival model. We first show that our positive results (i.e., lower bounds) for simple graphs extend to multigraphs. We then establish a stronger hardness result for multigraphs with respect to the optimal fractional solution.

We first claim that Theorem 3 holds also with respect to multigraphs. The proof holds almost intact, except that x_e adjusted accordingly. We present the proof here for completeness.

THEOREM 7. *There is a $\frac{1}{3}$-OCRS for matching in general multigraphs with edge arrivals.*

Proof. Let $c = \frac{1}{3}$. We prove that all $\alpha_e \leqslant 1$ for every edge e by induction on the set of edges, according to their arrival order. For the base case (an empty set), the argument holds trivially. We next prove the induction step. We can assume by the induction hypothesis that $\alpha_{e'} \leqslant 1$ for every edge $e' \in E$ but the last arriving edge $e = (uv)$. To finish the induction step we need to show that $\alpha_e \leqslant 1$. Recall that our algorithm matches each edge e' preceding $e = (uv)$ with probability $c \cdot x_{e'}$. Therefore,

$$\mathbf{Pr}\left[u \text{ is matched at } e = (uv)\right] = \sum_{e' < e : u \in e'} c \cdot x_{e'} \leqslant c \quad \text{and} \quad \mathbf{Pr}\left[v \text{ is matched at } e = (uv)\right] = \sum_{e' < e : v \in e'} c \cdot x_{e'} \leqslant c.$$

By the union bound, we have

$$\mathbf{Pr}\left[u, v \text{ are unmatched at } e = (uv)\right] \geqslant 1 - \mathbf{Pr}\left[u \text{ is matched at } e = (uv)\right] - \mathbf{Pr}\left[v \text{ is matched at } e = (uv)\right] \geqslant 1 - 2c.$$

For $c = 1/3$, $1 - 2c = c$. Thus,

$$\mathbf{Pr}\left[u, v \text{ are unmatched at } e = (uv)\right] \geqslant c \quad \text{and} \quad \alpha_{e=(uv)} = \frac{c}{\mathbf{Pr}[u, v \text{ are unmatched at } e = (uv)]} \leqslant 1,$$

as desired. This concludes the proof. □

We next claim that Lemma 1 holds with respect to multigraphs, and thus Theorem 4 as well. The proof of Lemma 1 holds almost as is, except that now there may be multiple edges between u and v. As a result, in Equation (16), we need to separate the edges into ones that are of the form (u, v) and ones that are not. For edges of the latter form, the analysis remains intact. Having multiple edges of the form (u, v) introduce positive correlation for u and v being matched simultaneously. Since our analysis works by establishing lower bound on the positive correlation for u and v being matched simultaneously, having multiple edges of the form (u, v) works in our favor.

We next present an improved upper bound for multigraphs (with respect to the optimal fractional solution).

PROPOSITION 3. *For multigraphs, under the edge arrival model, no online algorithm can get better than $\frac{2}{5}$ of f-OPT, even for 3-vertex multigraphs.*

Proof. Consider the graph depicted in Figure 3 with 3 vertices a, b, c, where edges (ab), (bc), (ac) have a deterministic weight of 1, and the second edge from a to b has a weight $\frac{1}{\epsilon}$ with probability ϵ (for an arbitrarily small ϵ), and 0 otherwise. We refer to the latter edge as the big edge. Suppose the three deterministic edges arrive first, followed by the big edge.

The optimal fractional solution is the following: if the big edge's weight is non-zero (this happens with probability ϵ), then take it; else, take each of the deterministic edges with probability 1/2. This gives approximately $\epsilon\frac{1}{\epsilon} + (1-\epsilon)\frac{3}{2} \approx \frac{5}{2}$. One can easily verify that no online algorithm can achieve more than weight 1. □

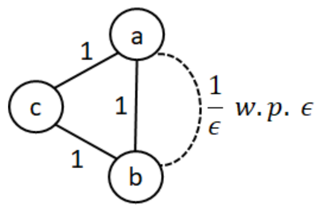

FIGURE 3. Upper bound for matching prophet inequality with edge arrival with respect to optimal fractional matching in multigraphs. Solid lines have weight 1; the dotted line has weight $1/\epsilon$ w.p. ϵ.