Single-Sample Prophet Inequalities via Greedy-Ordered Selection

Constantine Caramanis[*]   Paul Dütting[†]   Matthew Faw[*]   Federico Fusco[‡]

Philip Lazos[‡§]   Stefano Leonardi[‡]   Orestis Papadigenopoulos[*]

Emmanouil Pountourakis[¶]   Rebecca Reiffenhäuser[‡]

Abstract

We study *single-sample prophet inequalities* (SSPIs), i.e., prophet inequalities where only a single sample from each prior distribution is available. Besides a direct, and optimal, SSPI for the basic single choice problem [Rubinstein et al., 2020], most existing SSPI results were obtained via an elegant, but inherently lossy reduction to order-oblivious secretary (OOS) policies [Azar et al., 2014]. Motivated by this discrepancy, we develop an intuitive and versatile greedy-based technique that yields SSPIs *directly* rather than through the reduction to OOSs. Our results can be seen as generalizing and unifying a number of existing results in the area of prophet and secretary problems. Our algorithms significantly improve on the competitive guarantees for a number of interesting scenarios (including general matching with edge arrivals, bipartite matching with vertex arrivals, and certain matroids), and capture new settings (such as budget additive combinatorial auctions). Complementing our algorithmic results, we also consider mechanism design variants. Finally, we analyze the power and limitations of different SSPI approaches by providing a partial converse to the reduction from SSPI to OOS given by Azar et al.

1 Introduction

Prophet inequalities are fundamental models from optimal stopping theory. In the simplest version of prophet inequalities [Krengel and Sucheston, 1977, 1978, Samuel-Cahn, 1984], a set of n elements with rewards $r_1, r_2, \ldots, r_n$ sampled independently from known distributions $D_1, D_2, \ldots, D_n$ arrive in a fixed order. After each element is observed, the online algorithm can either accept the reward and stop, or drop it and move to consider the next element in the sequence. The goal is to show an α-competitive *prophet inequality*, i.e., an online algorithm such that the expected reward of the computed solution is at least a $1/\alpha$-fraction of the expected value of an optimal solution that can observe the values of all rewards before making a choice. The original work of Krengel and Sucheston [1977, 1978] and Samuel-Cahn [1984] has established a tight 2-competitive prophet inequality for the single-choice selection problem, i.e., the problem of selecting only one out of n elements.

Motivated also by applications to algorithmic mechanism design and to pricing goods in markets, later work has extended prophet inequalities to a broad range of problems, including matroids and polymatroids, bipartite and non-bipartite matching, and combinatorial auctions [e.g., Alaei, 2014, Chawla et al., 2010, Kleinberg and Weinberg, 2012, Dütting and Kleinberg, 2015, Feldman et al., 2015, Dütting et al., 2020a,b].

For many applications, the assumption of knowing the distributions exactly, as in the prophet model, is rather strong. It is consequential to ask whether near-optimal or constant-factor prophet inequalities are also achievable with limited information about the distributions. A natural approach in this context (pioneered by Azar et al. [2014]) is to assume that the online algorithm has access to a limited number of *samples* from the underlying distributions, e.g., from historical data. Arguably, a minimal assumption under this approach is that the online algorithm has access to just a *single* sample from each distribution.

Surprisingly, even under this very restrictive assumption, strong positive results are possible. Importantly, for the classic single-choice problem one can obtain a factor 2-approximate prophet inequality with just a single sample

---

[*]The University of Texas at Austin, USA. Email: {`constantine,matthewfaw`}`@utexas.edu`, `papadig@cs.utexas.edu`

[†]Google Research, Zürich, Switzerland. Email: `duetting@google.com`

[‡]Sapienza University of Rome, Italy. Email: {`fuscof,plazos,leonardi,rebeccar`}`@diag.uniroma1.it`

[§]IOHK. Email: `philip.lazos@iohk.io`

[¶]Drexel University, Philadelphia, USA. Email: `manolis@drexel.edu`

[Rubinstein et al., 2020], matching the best possible guarantee with full knowledge of the different distributions. The same problem, but with identical distributions, was considered by Correa et al. [2019], who showed that with one sample from each distribution it is possible to achieve a $e/(e-1)$ guarantee. This bound was subsequently improved by Kaplan et al. [2020] and Correa et al. [2020a, 2021, 2020b].

The central question we study in our work is whether similar results can also be obtained for more general combinatorial settings. This question was investigated first by Azar et al. [2014], who showed that for some combinatorial problems it is possible to achieve the same asymptotic guarantees as with full knowledge of the distributions with just a single sample from each distribution. The results they give are largely based on an elegant reduction to *order-oblivious secretary* (OOS) algorithms: in this model, there is no sample or other information about the distributions available - instead, a constant and random fraction of the input is *observed* for statistical reasons. Then, only on the remainder of the instance, an adversarial-order online algorithm is run to select elements. The reduction of SSPIs to OOSs relies on the following observation: with the single sample available to an SSPI, one can replace the sampled part of the instance in the first, statistical phase of any OOS. Since the sample values are drawn from the exact same distributions as those that were part of the original problem instance, this results in the exact same input distribution as the original OOS (and, therefore, the same competitive guarantee).

However, SSPIs derived from OOSs in this fashion are inherently lossful: the information from the sample available to such an SSPI remains partly unused, since the secretary algorithm utilizes only those sampled values that belong to elements in its statistical, not its selection phase. In addition, the elements used in the statistical phase are *automatically* rejected. We deviate from designing SSPIs via this construction and instead give a versatile and intuitive technique to design SSPIs *directly*. Our focus is to design SSPIs directly for matchings, matroids, and combinatorial auctions that are central to the area of secretary and prophet algorithms.

Besides presenting improved or completely new SSPIs for many such problems, we also investigate the power and limitations of the SSPI paradigm with respect to prior-free OOSs on one side, and full-information prophet inequalities on the other. Here, we take important steps towards solving the central open question whether constant-factor SSPIs exist for the prominent, but notorious problems of combinatorial auctions and general matroids.

1.1 A New Framework for Single-Sample Prophet Inequalities Our first contribution is a framework for deriving SSPIs *directly* via greedy-ordered selection algorithms as we describe in the following. This framework can be applied to a range of problems in the SSPI setting, two of which we put special focus on: First, we derive SSPIs with considerably improved competitive ratios for variants of max-weight matching problems on general/bipartite graphs, extending also to strategic settings. Second, we employ a richer version of the same design principles to provide the first constant-factor SSPI for a combinatorial auction setting, namely combinatorial auctions with *budget-additive* buyers. Note that we are able to obtain especially the latter result only via exploiting the high abstraction level and clear structure of probabilistic events within our framework.

1.1.1 Single-Sample Prophet Inequalities via Greedy-Ordered Selection Consider the general problem of selecting online a feasible subset from a number of elements, which arrive in some adversarial order. We propose a very general and versatile technique for selecting elements in our online algorithms that can be seen as a generalization of several approaches in the literature.

At its core, our method is based on two basic and simple insights. As a first ingredient, we use the well-known greedy algorithm, which considers all elements of a set decreasingly by their value and selects an element if and only if the resulting set of chosen elements is still feasible. The greedy algorithm, although it is offline in the sorting phase, behaves like an online algorithm on the ordered set of elements. Our second ingredient is the fact that drawing two values r_e, s_e from the distribution of an element e is equivalent to drawing $v_{e,1}, v_{e,2}$ and then flipping a fair coin to decide which one to use as the realization of r_e, and which as the sample s_e.

On a high level, our algorithms work as follows:

Greedy Phase: first, run a greedy procedure on the sampled values of all elements, and remember for every element e the threshold-value τ_e for which the following holds: element e would have been selected by greedy if we added it to the set of samples with a value $r_e > \tau_e$.

Online Phase: second, on arrival of an element in the adversarial order of the SSPI, observe its reward r_e. If $r_e > \tau_e$, and e together with all so-far selected elements is still feasible, collect e.

The reason that variations of this simple algorithmic idea yield constant-factor approximations is a combination

of our two observations from above. First, the *online* character of greedy after sorting all elements ensures that whether any element is chosen only depends on the *larger* values considered before, but is independent from any later steps, which we heavily rely on in our proofs. Second, the described process of first drawing *two* values from each element's distribution and only later deciding which is the actual reward relates the quality of the greedy solution on the samples to the online selection process on the actual rewards, and ensures the described thresholds are feasible.

As mentioned before, this type of idea is employed somewhat differently in many existing works. For example, and most closely related to our work, the optimal SSPI of [Rubinstein et al., 2020] for the single-choice setting can be interpreted as an implementation of the same paradigm. Also closely related is work by Korula and Pál [2009], who use greedy pricing combined with the random process of assigning values to the statistics or selection phase of a secretary algorithm (instead of sample or reward value in a SSPI) for bipartite weighted matching, and Ma et al. [2016], who apply the greedy idea to submodular secretary problems.

We provide a general and abstract view towards methods of the above type that aids the proofs of our approximation results by making the two key insights presented above very explicit. First, we use the online-offline-properties of greedy to state an *equivalent offline version* to each of our SSPIs, enabling us to much more easily keep track of probabilistic events and the computed thresholds. Second, we use the view of deciding only in the very last moment which of the two values drawn for an element is the sample or the reward to define (and bound the size of) a set of *safe* elements which even the adversarial arrival order cannot prevent us from picking.

Note that the above abstraction is indeed crucial for obtaining our results, since compared to previous work, we face some extra challenges. First, our problems are combinatorially richer than, e.g., the single-choice prophet inequality, requiring careful and dedicated adjustments to the greedy algorithms and thresholds we use for the first algorithm phase. Second, as opposed to applications in the secretary paradigm, the arrival orders in our model are fully adversarial, which poses an extra obstacle to guaranteeing a *safe* set in the above sense. Finally, the fact that (again opposed to secretary algorithms) we deal with more than one value associated with the same element (sample and reward) breaks the convenient independence of some of the greedy algorithm's decisions. This requires delicate handling of the resulting dependencies.

1.1.2 New and Improved Single-Sample Prophet Inequalities

We now present the results we derive within our greedy framework, which allows us to significantly improve on the competitive guarantees of many existing SSPIs, and provide the first constant-factor SSPIs for some problems previously not covered by this paradigm.

Max-weight matching with general edge arrivals. In Section 2, we apply the above ideas to the case of maximum weight matching in general (non-bipartite) graphs with edge arrivals – Here, edges $e \in E$ of a graph $\mathcal{G}$, each associated with a certain nonnegative weight $w(e)$, arrive one by one in adversarial order. The algorithms aims at selecting a maximum-weight matching $M \subseteq E$, such that each vertex has at most one incident edge in M. Before our work, the state-of-the-art for this problem was a 512-competitive algorithm derived via Azar et al. [2014] and the 256-competitive OOS in Feldman et al. [2016] for bipartite graphs, by choosing a random bipartition. We improve this to a 16-competitive SSPI derived via our framework. The proof of this result is exemplary for the key insights that drive our analyses.

Max-weight bipartite matching. In Section 3, we focus on the case of bipartite matching with vertex arrival where the items are available offline, and buyers arrive in adversarial order. Here, we provide an 8-competitve SSPI, again improving considerably over the 256-competitive SSPI.[1] The improved factor is obtained by exploiting the fact that due to vertex arrival, all edges of the same buyer become available at once. Then, we complement this result with a truthful mechanism, which is based on the idea of tightening the threshold requirements for each edge in a way that agents cannot hurt the approximation too much by picking the *wrong* elements on arrival. In particular, we give a 16-approximate mechanism, which to the best of our knowledge is the first single-sample result for this setting. Finally, we show an 8-competitive SSPI for the closely related (and somewhat more restricted) problem of selecting an independent set of elements in a transversal matroid, improving on the previous 16-approximation by Azar et al. [2014].

---

[1]We observe however that a SSPI with better competitive ratio for this problem can also be achieved by adjusting the analysis in Korula and Pál [2009] to become order-oblivious, which yields a 13.5 competitive policy.

Combinatorial set	Previous best	Our results
General matching (edge arrivals)	512	16
Budget-additive combinatorial auction	N/A	24
Bipartite matching (edge arrivals)	256	16
	6.75 (degree-d)	16 (any degree)
	$\mathcal{O}(d^2)$-samples	1 sample
Bipartite matching (vertex arrivals)	13.5	8
Transversal matroid	16	8
Graphic matroid	8	4
Co-graphic matroid	12	6
Low density matroid	$4\gamma(M)^3$	$2\gamma(M)$
Column k-sparse linear matroid	$4k$	$2k$

Table 1: Summary of main results

Budget-additive combinatorial auctions. In Section 4 we consider a more general problem, where buyers can be allocated more than one item, and each buyer b's valuation is of the form $v_b(S) = \min\{\sum_{i \in S} v_b(i), C_b\}$, for S a set of items and C_b a publicly known, buyer-specific budget. Here, a buyer's valuation for being assigned additional items changes with the assignment of any previous ones. This dependence, although it makes the design and, to an even greater extent, analysis of our algorithms considerably more involved, can still be resolved without incurring too much loss. As a result, we are able to show a 24-competitive SSPI for this model. This, to the best of our knowledge, is the first SSPI for a combinatorial auction setting beyond the simple case of additive valuations, and raises hope that the scope of constant-factor SSPIs reaches even further towards that of full-information prophet inequalities.

Matroids. In Section 5, we further demonstrate the improvement potential of directly designing SSPIs by considering matroids which satisfy a natural partition property. As we observe, many OOS policies for matroids rely on decomposing the given matroid into several *parallel* (in terms of feasibility) rank-1 instances, and then run an OOS policy (in parallel) on each of these instances. Rather than using the 4-competitive single-choice OOS policy of Azar et al. [2014], the same partitioning allows us to leverage the results of Rubinstein et al. [2020], which give a 2-competitive policy for the SSPI problem on rank-1 matroids. This allows us to improve the competitive guarantees for almost all the matroids considered in Azar et al. [2014] by a factor of 2.

Summary of our results and additional properties. In Table 1, we outline our improved SSPIs and compare them to previous work. As we have already mentioned, almost all existing SSPIs follow from the reduction to OOS, due to Azar et al. [2014], in combination with existing OOS policies (see [Dimitrov and Plaxton, 2012] for transversal matroid, [Korula and Pál, 2009] for graphic matroid and bipartite matching with vertex arrivals, and [Soto, 2013] for co-graphic, low density, and column k-sparse linear matroids).

We remark that (with the exception of budget-additive combinatorial auctions) all of our algorithms are *ordinal* (also known as *comparison-based*), in the sense that they do not require accurate knowledge of the values of the random variables, but only the ability to compare any two of them. Further, we emphasize that our competitive guarantees hold against an *almighty fully-adaptive adversary* – that is, an adversary who is a priori aware of *all* sample/reward realizations, and can decide on the arrival order of the elements in an adaptive adversarial manner.

Finally, we remark that our improved SSPIs can be used to provide truthful mechanisms with improved approximation guarantees comparing to Azar et al. [2014] for welfare and revenue maximization.[2] We refer the reader to Azar et al. [2014] for more details.

1.2 Limitations of Single-Sample Prophet Inequalities We have seen that for a large range of problems, SSPIs provide guarantees close to those of traditional prophet inequalities, and *direct* SSPIs as in our results improve consistently on those derived via Azar et al. [2014]'s reduction to OOSs. This positions the SSPI paradigm

---

[2]Note that, for the case of revenue maximization, and since our algorithms make use of all the available samples, an extra sample is needed to be used as *"lazy reserves"* (see Azar et al. [2014] for a definition).

[3]The *density* of a matroid $M(E, \mathcal{I})$ is defined as $\gamma(M) = \max_{S \subseteq E} \frac{|S|}{r(S)}$, where r is the rank function.

in between full-information prophets and the prior-free OOSs. In consequence, *direct* SSPIs raise hope for problems that offer good prophet inequalities, but remain resistant to constant-factor OOSs. Here, while we are making some progress with respect to combinatorial auction settings by providing a constant-factor SSPI for the budget-additive case, our results on matroids – although improving in terms of competitive ratio – do not extend to settings beyond those covered by constant-factor OOSs. This limitation to our progress is, as we observe, not a coincidence. In fact, we analyze more generally the power of SSPIs versus that of OOSs, and show the following partial reverse to the reduction of Azar et al. [2014]:

THEOREM. (INFORMAL) *For a large and intuitive subclass of SSPIs, which we call* pointwise *or P-SSPIs, the existence of a constant-factor P-SSPI policy implies that of an according OOS policy.*

This shows that with our, and all other existing SSPI methods, there is no hope to achieve constant-factor SSPIs for problems that do not allow for constant-factor OOSs. E.g., if any of the existing SSPI strategies did yield a constant approximation to the general matroid prophet inequality problem, this would also solve the famous matroid secretary conjecture of Babaioff et al. [2007] [4].

1.3 Further Related Work A great deal of attention has been given to multi-choice prophet inequalities under combinatorial constraints such as uniform matroid constraints [Hajiaghayi et al., 2007, Alaei, 2014] and general matroid constraints [Chawla et al., 2010, Kleinberg and Weinberg, 2012, Feldman et al., 2016]. A number of works have obtained prophet inequalities for matchings and combinatorial auctions [Alaei et al., 2012, Feldman et al., 2014, Ehsani et al., 2018, Gravin and Wang, 2019, Dütting et al., 2020a,b, Ezra et al., 2020]. In particular, Gravin and Wang [2019] provide prophet inequalities for weighted bipartite matching environments under edge arrivals, and show a lower bound of 2.25 on the competitive ratio of any online policy for this setting, proving that this problem is strictly harder than the matroid prophet inequality. Quite recently, Ezra et al. [2020] provided a near-optimal prophet inequality for general weighted *(non-bipartite) matching* under edge arrivals. Beyond these settings, recent work has considered the prophet inequality problem under arbitrary packing constraints [Rubinstein, 2016, Rubinstein and Singla, 2017].

Regarding the prophet inequality problem in the limited information regime, in addition to their meta-result connecting SSPIs with OOS algorithms, Azar et al. [2014] provide a threshold-based $(1 - \mathcal{O}(1/\sqrt{k}))$-competitive algorithm for k-uniform matroids. Further, Rubinstein et al. [2020] develop a $(0.745 - \mathcal{O}(\epsilon))$-competitive algorithm for the single-choice IID case, using $\mathcal{O}(n/\epsilon^6)$ samples from the distribution, improving on the results of Correa et al. [2019]. Prophet inequalities that require a polynomial number of samples can also be obtained via the balanced prices framework [Feldman et al., 2015, Dütting et al., 2020a].

The motivation for studying the prophet inequality problem in many of the above settings comes largely from connections to mechanism design. While optimal single-parameter mechanisms for both welfare [Vickrey, 1961, Groves, 1973, Clarke, 1971] and revenue [Myerson, 1981] have been well-understood for decades, there has been an extended study of simple and practical mechanisms that approximate these objectives. Prophet inequalities are known to be a powerful tool for designing simple posted-price mechanisms. Azar et al. [2014] show how to apply results of Chawla et al. [2010], Azar et al. [2013], Dhangwatnotai et al. [2010] to obtain mechanisms in settings where the mechanism designer only has access to a single sample (or a constant number of samples) from the distribution of the agents' values. A recent work of Dütting et al. [2021] also investigates single-sample mechanism design for two-sided markets. The authors prove matching upper and lower bounds on the best approximation that can be obtained with one single sample for subadditive buyers and additive sellers. Moreover, computationally efficient blackbox reductions that turn any one-sided mechanism into a two-sided mechanism with a small loss in the approximation, while using only one single sample from each seller are provided.

In concurrent and independent work, a greedy technique similar to our framework was implemented for the problem of online weighted matching by Kaplan et al. [2021]. Their main model, adversarial-order model with a sample, differs from ours in that it is closer to secretary algorithms than SSPI: instead of having two values for each element, reward and sample, they assume the bipartite graph to consist of a *historic* part that is available to the algorithm, and the actual problem instance. Using an intermediate step, the so-called two-faced model, they derive constant factor approximation for SSPI for bipartite and general matching. In particular, they show a

---

[4]We further remark that our reduction for P-SSPI, together with our P-SSPI for budget-additive combinatorial auction, also implies a constant-factor OOS for this problem.

13.5-approximation for the general graph, edge arrival model and a $3 + 2\sqrt{2} \approx 5.83$-approximation for the bipartite, vertex arrival model.

1.4 Organization of the Paper We start by implementing our framework for the problem of max-weight matching in general graphs with edge arrivals in Section 2, which offers the most straightforward version of our online-offline analyses. In Section 3, we present our results on bipartite matchings with one-sided vertex arrival, as well as their strategic/mechanism variant (Section 3.2) and the related problem of transversal matroids (Section 3.3). In Section 4, we move on to the problem of combinatorial auctions with budget-additive buyers, whose algorithm and proofs build on the previous results, but are considerably more challenging. In Section 5, we strengthen our observation that *direct* SSPIs offer better competitive factors than those derived from OOSs: in addition to our previous improvements on the results obtained for matching problems via Azar et al. [2014]'s reduction, we do the same also for a wide class of matroid problems captured by such SSPIs. Finally, on the negative side, we show via a partial reverse to Azar et al. [2014]'s reduction in Section 6 that for a large class of SSPIs, the improvement over OOS policies is indeed limited to a constant. All omitted proofs can be found in Appendix A.

2 Edge Arrival in General Graphs

In this section, we consider the case of matching on a general graph $\mathcal{G} = (V, E)$ with edge arrivals, where the weight of each edge $e \in E$ is drawn independently from a distribution D_e. We recall that, for the offline problem where the edge weights are known a priori, the greedy algorithm that collects edges in a non-increasing order of weight while maintaining feasibility is a 2-approximation. Our algorithm for this problem outlines very well the idea we described above: to bridge the gap between the fixed order employed by greedy and the adversarial one of prophet inequalities (where greedily adding edges would be arbitrarily bad), we utilize the independent single samples $S = \{s_e \mid e \in E\}$ that we have from all the edge distributions.

Note that throughout, we consider *weighted sets* to denote our edge sets and allocations, in the sense that they contain for each edge e also the information about its associated weight $w(e)$. Further, we assume to break ties uniformly at random before running the algorithm, i.e. for the set of all drawn numbers that have value x, pick a permutation π of those uniformly at random and define a draw of the same value to be larger than another if and only if it comes first in π.

The algorithm first computes an *offline* greedy matching M_S on the graph with edge weights S. Then, for each vertex v, it interprets the weight of the edge incident to it in the greedy solution as a *price* p_v for the adversarial-order online algorithm, with the convention that $p_v = 0$ if there is no edge in M_S incident to v. In the online phase, whenever an edge $e = \{u, v\}$ arrives and both endpoints are free (i.e., not already matched), it is added to the matching M if and only if $r_e \geq \max\{p_v, p_u\}$. To facilitate the analysis, together with the solution M, the algorithm builds E', a subset of E containing all the edges such that $r_e \geq \max\{p_v, p_u\}$, i.e., all edges that are price-feasible. Note that E' is a superset of the actual solution M and may not be a matching. See Algorithm 1 for a pseudocode.

While the above algorithm is simple, the analysis is more challenging. As pointed out before, the greedy method can be seen as an offline algorithm as well as an online algorithm with the extremely simplifying assumption of a fixed and weight-decreasing arrival order. We combine such a shift of viewpoint with a suitable reformulation of the random processes involved. More precisely, we look at our (online) algorithm as the byproduct of a run of a (fixed-order!) greedy strategy on a random part of a fictitious graph that contains each edge *twice*: once with the true weight r_e, and once with the sampled weight s_e.

Equivalent offline algorithm. To analyze the competitive ratio of Algorithm 1, we state an offline, greedy-based algorithm with similar properties that is easier to work with. This offline procedure considers an equivalent stochastic process generating the edge weights in S and R. Instead of first drawing all the samples, and then all the realized weights, we consider a first stage in which, for each edge e, two realizations from D_e are drawn. Then, all drawn realizations ($2|E|$ in total) are sorted in decreasing order according to their values, breaking ties at random as explained above. Finally, the offline algorithm goes through all these values in that order; each value is then associated with S or R by a random toss of an unbiased coin. Consequently, the second, smaller weight of the same edge arriving later is associated to the remaining category with probability 1. In particular, an edge e is marked as R-used, when $r_e > s_e$, and S-used, otherwise. It is not hard to see that the distributions of S and R are identical for both random processes.

Algorithm 1: Prophet Matching with Edge-Arrivals

Set $E' = \emptyset$, $M = \emptyset$

Compute the greedy matching M_S on the graph with edge weights according to sample S

for *each* $e = \{u, v\} \in M_S$ **do**

    Set $p_u = p_v = s_e$

for *each vertex* $u \in V$ *unmatched in* M_S **do**

    Set $p_u = 0$

for *each arriving* $e = \{u, v\} \in E$ **do**

    if $r_e \geq \max\{p_u, p_v\}$ **then**

        Add e to E'. `// This is just for the analysis.`

        if u, v *are not matched in* M **then**

            Add e to M with weight r_e

return M

The pseudocode for this offline procedure is given in Algorithm 2. As already mentioned, it mimics in an offline way the construction of M_S and E' of Algorithm 1, as shown in the following.

Algorithm 2: Offline Simulation for Matching with Edge-Arrivals

---

Set $E' = \emptyset$, $M_S = \emptyset$, $M = \emptyset$, and $V^S = V$

For each $e \in E$, draw from D_e two values $a_{e,1}$ and $a_{e,2}$

Order $A = \{a_{e,1}, a_{e,2} | e \in E\}$ in decreasing fashion

for *each value* $a \in A$ *in the above order* **do**

    if *a corresponds to edge* $e = \{u, v\}$ *that has never been observed before* **then**

        Flip a coin

        if *Heads* **then**

            Mark e as R-*used*

            if $u \in V^S$ *and* $v \in V^S$ **then**

                Add e to E' with weight a

        else

            Mark e as S-*used*

            if $u \in V^S$ *and* $v \in V^S$ **then**

                Add e to M_S with weight a

                Remove u and v from V^S

    else if *a corresponds to edge* $e = \{u, v\}$ *which is R-used,* $u \in V^S$ *and* $v \in V^S$ **then**

        Add e to M_S with weight a

        Remove u and v from V^S

for *each* e *in the same order as Algorithm 1* **do**

    if e *in* E' *and* $\{e\} \cup M$ *is a matching* **then**

        Add e to M with weight r_e

return M.

CLAIM 2.1. *The sets E', M_S and M are distributed in the same way when computed by Algorithm 1 as when computed by Algorithm 2.*

Proof. As a preliminary result, we claim that the distribution of all rewards in R and samples in S is exactly the same in both Algorithm 1 and Algorithm 2. This follows by the fact that, for each edge e, the distribution of r_e remains the same if we simply sample r_e from D_e, independently, or if we sample two values $a_{e,1}$ and $a_{e,2}$ from D_e, independently, and then set r_e equal to one of the two equiprobably. The symmetric argument holds for any sample value in S.

Now, we just need to show that for *any* realization of the S and R values, the corresponding sets of edges of interest are exactly the same in the offline and online setting. To avoid complication, we deterministically use the same tie breaking rules in both the online and offline case.

First, notice that if we restrict the offline algorithm to consider only the samples S, we recover exactly the procedure to generate a greedy matching with respect to the samples. This is true because what happens to the values in R has no influence on M_S. Hence, the two versions of M_S in the different algorithms indeed follow the same distribution. In fact, this is even true step-wise: at any point in time during Algorithm 2, M_S contains the greedy matching on samples *restricted* to the edges whose s_e is greater than the value being considered at that specific time.

We conclude the proof by showing that the two versions of E' coincide. Once we have that, it is enough to observe that given the same fixed arrival order, the matchings M are extracted in the same way from E'. To this end, consider an iteration of Algorithm 1 in which an edge e arrives in the online phase. We argue that e is added to E' in the offline procedure if and only if the online procedure does the same. Indeed, fix edge $e = \{u, v\}$ added to E' by Algorithm 1. Then, the online version of M_S contains no edges incident to u or v with larger sample than r_e, because this would imply a price higher than r_e in at least one of the two endpoints. Since the M_S are the same in both algorithms, when Algorithm 2 considers e, it holds u, $v \in V^S$ (u and v are not yet matched in M_S) and Algorithm 2 will also add e. The other direction follows analogously. $\square$

Given the above result, we use in the following E', M_S and M mainly to refer to Algorithm 2, but the same would apply to Algorithm 1.

Correctness and competitive analysis. The correctness of Algorithm 1 is guaranteed by the fact that the algorithm never adds to M any edge incident to an already matched vertex. Thus, the set of collected edges in M is a valid matching.

We now proceed in analyzing the competitive guarantee of our algorithm. In particular, we show that Algorithm 1 is 16-competitive, namely, for any adversarial arrival order of the edges, the algorithm retains in expectation at least a 1/16-fraction of the expected value of optimal offline matching OPT. The high level idea of the proof lies in relating the optimal offline matching and our solution by using M_S (which yields a 2-approximation of OPT) and a carefully chosen subset of E', the *safe* edges.

DEFINITION 2.1. (SAFE EDGES) *Let $e = \{u, v\}$ be an edge incident to vertex v. We call e* safe *for v, if the following conditions are met:*

1. *e is the only edge in E' incident to v.*

2. *There is no edge in E' that is incident to u and has smaller weight than r_e.*

Let us denote e_v as the edge incident to v in E' of maximal reward (if such an edge exists in E'). We are able to show the following convenient property for this edge being safe for v:

LEMMA 2.1. *For any vertex v and edge e incident to v for which the probability that $e_v = e$ is non-zero, we have that*

$$\mathbb{P}[e_v \text{ is safe for } v \mid e_v = e] \geq \frac{1}{4}.$$

Proof. Recall than an edge is marked as R-used if its realized reward is greater than the corresponding sample. In the opposite case, the edge is marked as S-used. By construction of Algorithm 2, E' contains only R-used edges. Consider now the time where some edge $e_v = e = \{v, u\}$ with value a is added to set E'. By definition of e_v, this is the first edge added in E' that is incident to vertex v.

Consider now the next time where some edge e' of value a', incident to either u or v (or both), and such that both endpoints of e' are in V^S, is parsed by Algorithm 2. We distinguish among the following cases:

Case 1: $e' = \{v, u'\}$**, where** $u' \neq u$**.** In this scenario, we have that $e' = \{v, u'\}$, namely, e' is incident to vertex v but not u. We now show that, with probability at least 1/2, the first condition of Definition 2.1 is satisfied, by considering two sub-cases: **(a)** In the case where e' has been observed before (i.e., a' is its second realization), then e' is R-used and, thus, a' is a sample value. Indeed, if e' were S-used, then at least one of the its endpoints would no longer be in V^S, a fact which would contradict the choice of e'. Therefore, since e' is R-used and both its endpoints are in V^S by the time a' is parsed, vertex v (the only common vertex between e and e') is now removed from V^S and, thus, no other edge incident to v can be added in E'. Thus, e_v must be the only edge in E' incident to u, as desired. **(b)** In the case where e' has not been observed before by Algorithm 2, it is easy to see that, with probability 1/2, e' is S-used. Therefore, by the same argument as above, e_v is the only edge in E' incident to v.

Once the first condition of Definition 2.1 is satisfied, let $e'' = \{u, u''\}$ be the next edge of value $a'' < a' < a$ that is parsed by Algorithm 4, such that $u, u'' \in V^S$. By repeating the exact same arguments as in the previous paragraph, we can see that the second condition of Definition 2.1 will additionally be satisfied with probability at least $1/2$. Hence, we can see that the probability that e_v is safe for v is at least $1/4$.

Case 2: $e' = \{v', u\}$**, where** $v' \neq v$**.** By exactly the same arguments, the above analysis can be replicated, even if the first parsed edge after the time where e_v is added to E' is incident only to vertex u, but not v.

Case 3: $e' = \{v, u\}$**.** In this case, e' either corresponds to an edge parallel to e_v, or to edge e_v itself. If e' is a parallel edge to e_v, then, by an analysis similar to the first case, both conditions are satisfied with probability at least $1/2$. On the other hand, if e' coincides with e_v (which is by definition R-used), then it can be verified that e_v is already safe for v (with probability 1). □

Recall that e_v is defined to be the maximal-reward edge among the edges in E' incident to vertex v (if such an edge exists). We denote by r_{e_v} the reward of such an edge, and we set $r_{e_v} = 0$ if such an edge does not exist.

LEMMA 2.2. *For any realization of the edge weights in A, we have*

$$\mathbb{E}\left[\sum_{v \in V} r_{e_v} \mathbb{1}_{\{e_v \text{ is safe for } v\}}\right] \geq \frac{1}{4} \cdot \mathbb{E}\left[\sum_{v \in V} r_{e_v}\right].$$

Proof. Let us fix any realization of A and assume w.l.o.g. that for each edge $e \in E$ with realizations $a_{e,1}$ and $a_{e,2}$, it holds $a_{e,1} > a_{e,2}$. By linearity of expectations, in order to prove the Lemma, it suffices to show that for any vertex $v \in V$, it holds $\mathbb{E}\left[r_{e_v} \mathbb{1}_{\{e_v \text{ is safe for } v\}}\right] \geq \frac{1}{4} \cdot \mathbb{E}\left[r_{e_v}\right]$.

It is not hard to verify that for each edge $e \in E$ incident to vertex v, if e is safe for v, then $r_e = a_{e,1}$, the larger weight realization. Indeed, assuming that $r_e = a_{e,2}$, the value $a_{e,2}$ is parsed after $a_{e,1}$ by Algorithm 2 and, thus, it cannot be the case that $e \in E'$. Given this fact, we have

$$\mathbb{E}\left[r_{e_v} \mathbb{1}_{\{e_v \text{ is safe for } v\}}\right] = \sum_{e \in E} a_{e,1} \cdot \Pr\left[e_v \text{ is safe for } v \text{ and } e_v = e\right].$$

Now, by using the result of Lemma 2.1, we have that

$$\Pr\left[e_v \text{ is safe for } v \text{ and } e_v = e\right] = \Pr\left[e_v \text{ is safe for } v \mid e_v = e\right] \cdot \Pr\left[e_v = e\right] \geq \frac{1}{4} \cdot \Pr\left[e_v = e\right],$$

thus, concluding that

$$\mathbb{E}\left[r_{e_v} \mathbb{1}_{\{e_v \text{ is safe for } v\}}\right] \geq \frac{1}{4} \cdot \sum_{e \in E} a_{e,1} \cdot \Pr\left[e_v = e\right] \geq \frac{1}{4} \cdot \mathbb{E}\left[r_{e_v}\right].$$

The Lemma follows by summing the above equation over all vertices $v \in V$ and using linearity of expectation. □

LEMMA 2.3. *In any run of Algorithm 2, for the expected reward collected in M, we have*

$$\mathbb{E}\left[w(M)\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[\sum_{v \in V} r_{e_v} \mathbb{1}_{\{e_v \text{ is safe for } v\}}\right].$$

Proof. We first partition the set $M \subset E'$ into two sets M^B and M'. Informally, M^B is the subset of edges $e = \{u, v\}$ in M that are safe for both endpoints u and v, whereas M' contains the rest of the edges in M. Formally:

$$M^B = \{e = \{u, v\} \in M \mid e \text{ is safe for } u \text{ and for } v\} \quad \text{and} \quad M' = M \setminus M^B.$$

Similarly, we denote by V^B the subset of vertices that are adjacent to some edge in M^B, and by $V' = V \setminus V^B$ the rest of the vertices.

Recall that, by definition of safe edges, each edge $e = \{u, v\}$ in M^B is the only edge that can be collected by both endpoints u and v in the second phase of Algorithm 2. Further, since the weight of such an edge is contributes to both its endpoints, it follows directly that

$$w(M^B) = \frac{1}{2} \sum_{v \in V^B} r_{e_v} \mathbb{1}_{\{e_v \text{ is safe for } v\}}.$$

Turning now our attention to the set M', consider any vertex $v \in V'$ that is incident to some edge $e_v = \{v, u\}$ that is safe for v but not for u. By definition of the safe edges, by the time this edge arrives in the second phase of Algorithm 2, it is either collected by the algorithm, or another edge $e' = \{u, u'\}$, incident to u, has already been collected with reward $r_{e'}$. However, since e_v is safe for v, it has to be that $r_{e'} \geq r_{e_v}$, since no edge in E' incident to u has smaller reward than e_v. Notice that $e' = \{u, u'\}$ cannot be safe for u, given that in the above scenario E' contains at least *two* edges incident to u, which violates Definition 2.1. Further, e' cannot be safe for u' either, since E' contains e_v, which is incident to u and has reward *smaller* than $r_{e'}$, violating the second condition of Definition 2.1. Finally, adding any edge e to M' can make at most two safe edges infeasible: either the added edge e was safe itself, or after adding e, we can no longer include any safe edges incident to its endpoints, each of which has at most one safe, incident edge by definition. We can conclude that

$$w(M') \geq \frac{1}{2} \sum_{v \in V'} r_{e_v} \mathbb{1}_{\{e_v \text{ is safe for } v\}}.$$

By combining the above expressions, and since M^B and M' (resp., V^B and V') is a partition of M (resp., V). we get

$$w(M) = w(M^B) + w(M') \geq \frac{1}{2} \sum_{v \in V} r_{e_v} \mathbb{1}_{\{e_v \text{ is safe for } v\}}.$$

The proof of the Lemma follows simply by taking expectation over the above expression. □

THEOREM 2.1. *For the problem of finding a maximum-weight matching in a general graph $\mathcal{G}$, in the online edge arrival model, Algorithm 1 is 16-competitive in expectation, i.e.*

$$16 \cdot \mathbb{E}[w(M)] \geq \mathbb{E}[\mathsf{OPT}],$$

where OPT is the weight of an optimal matching in $\mathcal{G}$.

Proof. First, we claim that the set E' has expected weight at least $w(M_S)/2$. We define $\mathrm{val}_{\hat{E}}(v)$ for $v \in V$, and any *weighted* edge set $\hat{E} \subseteq E$ to denote the largest weight of an edge incident to vertex v in edge set $\hat{E}$, or 0 if there is no such edge. We fix any vertex $v \in V$ and consider the first time in the run of Algorithm 2, where an edge $e = \{v, u\}$ of value a arrives which is incident to v and, at that time, $u \in V^S$. Note that in the case where $u \notin V^S$, the edge cannot be added to neither E' nor M_s. By construction of the offline algorithm and since $v \in V^S$, with probability $1/2$ (i.e., the probability of the fair coin-flip), edge e is marked as R-used and it is added in E'. At this point during the run of Algorithm 2, no matter the outcome of the fair coin flip, at most one edge incident to v can be added to M_S – either e with weight a, or some other edge e' of weight *smaller* than a. Thus, for any $v \in V$, we have that $\mathbb{E}[\mathrm{val}_{E'}(v)] \geq \frac{1}{2} \cdot \mathbb{E}[\mathrm{val}_{M_S}(v)]$, which, by linearity of expectation implies that

$$\mathbb{E}\left[\sum_{v \in V} r_{e_v}\right] = \mathbb{E}\left[\sum_{v \in V} \mathrm{val}_{E'}(v)\right] \geq \frac{1}{2}\mathbb{E}\left[\sum_{v \in V} \mathrm{val}_{M_S}(v)\right] = \mathbb{E}[w(M_S)].$$

Now, by combining Lemmas 2.2 and 2.3, for the expected reward collected in M

$$\mathbb{E}[w(M)] \geq \frac{1}{2} \cdot \mathbb{E}\left[\sum_{v \in V} r_{e_v} \mathbb{1}_{\{e_v \text{ is safe for } v\}}\right] \geq \frac{1}{8} \cdot \mathbb{E}\left[\sum_{v \in V} r_{e_v}\right] \geq \frac{1}{8} \cdot \mathbb{E}[w(M_S)] \geq \frac{1}{16} \cdot \mathbb{E}[\mathsf{OPT}],$$

where in the last inequality we use the fact that $w(M_S)$ has the same distribution as the value of the greedy solution with respect to the rewards, which in turn is a 2-approximation of the optimal matching. □

3 Vertex Arrival in Bipartite Graphs

We now consider the case of SSPI matching on bipartite graphs with vertex arrivals. More specifically, we consider a bipartite graph $\mathcal{G} = (B \cup I, E)$, where B is the set of left vertices, which we refer to as "buyers", and I is the set of right vertices, which we refer to as "items". We remark that even though there is no incentive analysis in this section, a truthful version of our results, with slightly worse approximation guarantees, is presented in Section 3.2. Every edge $e = \{b, i\}$ between buyer $b \in B$ and item $i \in I$ is associated with an independent distribution D_e. In the offline phase, the gambler receives a single sample S from the product distribution of edge weights. In the online phase, buyers arrive sequentially and in adversarial order, while the gambler observes simultaneously the rewards of all edges incident to the arriving buyer. At each time, the gambler decides irrevocably whether to match the buyer to some unmatched item, or skip. The objective is to maximize the total expected reward collected by the buyers, against that of a prophet who can compute a max-weight matching w.r.t. the rewards R.

In this section, we first provide a $\mathcal{O}(1)$-competitive SSPI for the above model. Then, we show how our algorithm can be translated into a truthful mechanism where buyers are incentivized to report the true reward for each item. Finally, we show how our analysis extends to the case where the weights of the all the edges incident to the same buyer are identical (i.e., buyer-dependent), which allows us to provide an SSPI of improved competitive guarantee for the case of transversal matroids.

3.1 Main algorithm and analysis Our algorithm for this case operates in a similar manner as that of Section 2. As before, the sample S is used to calculate prices on both sides: items (fixed side vertices) have prices to *protect* them from being sold too cheaply, while buyers (incoming vertices) have prices they need to beat in order to participate in the market, in addition to beating the item prices.

Algorithm 3: Bipartite Prophet Matching with Vertex-Arrivals

Set $E^+ = \emptyset$, $M = \emptyset$
Compute a greedy matching M_S on the graph with edge weights $S = \{s_e \mid e \in E\}$
for *each* $e = \{b, i\} \in M_S$ **do**
    Set $p_i = p_b = s_e$
for *each vertex k not matched in M_S* **do**
    Set $p_k = 0$.
for *each arriving buyer $b \in B$* **do**
    Let $\hat{e} = \{b, i^*\} = \arg\max\{r_e \mid e = \{b, i\} \text{ and } r_e \geq \max\{p_b, p_i\}\}$
    Add $\hat{e}$ to E^+ with weight $r_{\hat{e}}$ // `Only used for the analysis.`
    if *i^* is not matched in M* **then**
        Add $\{b, i^*\}$ to M with weight $r_{\hat{e}}$
return M

We provide some intuition on the objects of interest that appear in the pseudo-code of Algorithm 3. We denote by M_S the greedy matching on the S graph, while by E^+ the set of edges of maximum reward (greater than their thresholds) for each arriving buyer $b \in B$. Let M be the matching returned by the algorithm on the edges of E^+, where conflicts are solved in favor of the first arriving edge. Note that M is computed in an online, adversarial ordering.

Equivalent offline algorithm. In order to relate the weight of E^+ and M with the prophet's expected reward, similarly to the previous section, we consider an offline version of the algorithm, which exhibits the same distribution over the relevant sets, and can be analyzed more easily. Indeed, Algorithm 4 interleaves the building of a greedy matching M_S with that of the allocation E^+. While all the values are drawn in advance, they are assigned to S and R only upon arrival. Using similar arguments as in the proof of Claim 2.1, we establish the following equivalence:

CLAIM 3.1. *The sets E^+, M_S and M are distributed the same way when computed by Algorithm 3 as when computed by Algorithm 4.*

Correctness and competitive analysis. The correctness of Algorithm 3 is guaranteed since only edges in E^+ can be added in M, E^+ contains at most one edge for each buyer, and the algorithm never assigns an item to more than one buyers. Hence, the set of collected edges in M is a valid matching.

We now analyze the competitive guarantee of our algorithm. In particular, we show that Algorithm 3 is 8-competitive. As in Section 2, the main idea of the proof is to relate the weight of the optimal offline matching, M_S, with the matching collected by our algorithm, M, using a carefully chosen subset of E^+, the *safe* edges.

DEFINITION 3.1. *We call an edge $e = \{b, i\} \in E$ safe for buyer b if the following conditions are true:*

1. *e is the only edge in E^+ incident to buyer b.*

2. *There is no edge in E^+ that is incident to item i and has smaller weight than r_e.*

As in Section 2, we denote $e_b \in E^+$ the edge incident to buyer b, among those in E^+, if such an edge exists, in a run of Algorithm 4. Additionally, we fix a realization of A, $b \in B$, and $e \in E$ for which the probability that $e_b = e$ is non-zero.

Algorithm 4: Offline Simulation for Bipartite Matching with Vertex-Arrivals

---

Set $E^+ = \emptyset$, $M_S = \emptyset$, $M = \emptyset$, $B^S = B^R = B$, and $I^S = I$
For each $e \in E$, draw from D_e two values $a_{e,1}$ and $a_{e,2}$
Order $A = \{a_{e,1}, a_{e,2} \mid e \in E\}$ in a decreasing fashion
for *each value $a \in A$ in the above order* **do**
    if *a corresponds to edge $e = \{b, i\}$ that has never been observed before* **then**
        Flip a fair coin
        if *Heads* **then**
            Mark e as R-used
            if *$b \in B^R$ and $i \in I^S$* **then**
                Add e to E^+ with weight a
                Remove b from B^R
        else
            Mark e as S-used
            if *$b \in B^S$ and $i \in I^S$* **then**
                Add e to M_S with weight a
                Remove b from B^S and B^R, and i from I^S
    else if *a corresponds to an edge $e = \{b, i\}$ which is R-used, $b \in B^S$ and $i \in I^S$* **then**
        Add e to M_S with weight a
        Remove b from B^S and i from I^S
for *each e in the same order as Algorithm 3* **do**
    if *$e \in E^+$ and $\{e\} \cup M$ is a matching* **then**
        Add e to M with weight r_e
return M.

LEMMA 3.1. *For any buyer $b \in B$ and edge e incident to b for which the probability that $e_b = e$ is non-zero, we have that*

$$\Pr\left[e_b \text{ is safe for } b \mid e_b = e\right] \geq \frac{1}{2}.$$

Proof. As in the proof of Lemma 2.1, we note that an edge is marked R-used if its realized reward is greater than the corresponding sample, and otherwise is marked as S-used. By construction, Algorithm 4 only adds R-used edges to E^+. Let us consider the time in the greedy ordering when $e_b = e = \{b, i\}$ is added to set E^+. By definition of e_b, this is the *first* time an edge incident to buyer b is added to E^+. Additionally, by construction of the algorithm, this must be the *only* edge corresponding to buyer b which is added to E^+. Thus, the first condition of Definition 3.1 is satisfied by construction.

Now, let us consider the next time in the greedy ordering when an edge $e' = \{b', i\}$ with weight a' is parsed by the algorithm, $b' \in B^S$, and $i \in I^S$ (indeed, if $b' \notin B^S$ or $i \notin I^S$, then e' could not be added to either E^+ or M_S).

If this is the second time that e' has been processed by the algorithm, then it must be the case that e' is R-used (since, otherwise, we get a contradiction to the choice of e_b), and hence, a' is a sample value. At this time, b' is removed from B^S and i from I^S, so no other edges incident to item i can be added to E^+ after this point. Thus, in this case, both conditions of Definition 3.1 are satisfied. Now, in the case when e' is being processed for the *first* time, we have that, with probability $1/2$, e' is S-used, thus guaranteeing that, by similar arguments as above, both conditions of Definition 3.1 are satisfied. □

LEMMA 3.2. *For any fixed realization of edge weights A, we have*

$$\mathbb{E}\left[\sum_{b\in B} r_{e_b}\mathbb{1}_{\{e_b \text{ is safe for } b\}}\right] \geq \frac{1}{2}\cdot\mathbb{E}\left[\sum_{b\in B} r_{e_b}\right].$$

Proof. The proof follows by the exact same arguments presented in Lemma 2.2, replacing the application of Lemma 2.1 with Lemma 3.1. □

LEMMA 3.3. *In any run of Algorithm 4, for the expected reward collected in M, we have*

$$\mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\sum_{b\in B} r_{e_b}\mathbb{1}_{\{e_b \text{ is safe for } b\}}\right].$$

Proof. The proof proceeds in a similar fashion to that of Lemma 2.3 for the case of general matching with edge-arrivals. Consider any fixed buyer b, and suppose that the edge $e_b = \{b, i\}$ is safe for b. We prove that either Algorithm 3 collects r_{e_b}, or there exists some other buyer b' and edge $e' = \{b', i\}$ collected by Algorithm 3 such that (i) $e' = e_{b'}$, (ii) $r_{e'} > r_{e_b}$, and (iii) $e_{b'}$ *cannot* be safe for buyer b'. The above three conditions, combined with the fact that, by definition, Algorithm 3 only collects edges in E^+, and that each buyer can preclude the collection of at most one safe edge, imply the following stronger version of the claimed inequality:

$$w(M) \geq \sum_{b\in B} r_{e_b}\mathbb{1}_{\{e_b \text{ is safe for } b\}}.$$

To prove (i), we simply note that, by definition of Algorithm 3, for each buyer b', either $e_{b'}$ is collected, or *no* edge adjacent to b' is collected. Claim (ii) follows by Definition 3.1 and the fact that only edges in E^+ can be collected by Algorithm 3. To establish (iii), observe that, by Definition 3.1, edges e' and e_b cannot be simultaneously safe, as they are incident to the same item i. □

THEOREM 3.1. *For the problem of finding a maximum-weight matching in a bipartite graph $\mathcal{G}$ in the online vertex arrival model, Algorithm 3 is 8-competitive in expectation, i.e.,*

$$8\cdot\mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\mathsf{OPT}\right],$$

where OPT is the weight of an optimal matching in $\mathcal{G}$.

Proof. First, observe that, by following the exact same arguments as in Theorem 2.1, $\mathbb{E}\left[w(E^+)\right] \geq 1/2\cdot\mathbb{E}\left[w(M_S)\right] \geq 1/4\cdot\mathbb{E}\left[\mathsf{OPT}\right]$. Hence, combining Lemmas 3.2 and 3.3,

$$\mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\sum_{b\in B} r_{e_b}\mathbb{1}_{\{e_b \text{ is safe for } b\}}\right] \geq \frac{1}{2}\cdot\mathbb{E}\left[\sum_{b\in B} r_{e_b}\right] = \frac{1}{2}\cdot\mathbb{E}\left[w(E^+)\right] \geq \frac{1}{8}\cdot\mathbb{E}\left[\mathsf{OPT}\right].$$

□

3.2 Truthful Bipartite Matching For the same model as above, with buyers on the left side of $\mathcal{G}$ arriving online and items on the right side available offline, we design an incentive-compatible version of our prophet inequality. The difficulty is that our 8-approximation in Theorem 3.1 relies on each buyer choosing the *largest* incident edge that beats an item's price. However, this item need not correspond to the item of highest *utility* to the buyer. Indeed, the price for this item might be much closer to the buyer's value than for other items. In this

case, a buyer might prefer to purchase an item which she values significantly less, in order to maximize her utility. For this reason, we cannot expect to show *any* competitive guarantee for the utility of the returned solution.

Clearly, a different approach is needed. We use a pricing-based algorithm that does not rely on restricting the buyer's choice of item (among those with feasible prices that are still available). Instead, we ensure a sufficient weight for the resulting matching solely via appropriate thresholds for each buyer and item: this can be seen as doing a worst-case analysis over all threshold-feasible edges instead of choosing a good subset of them, explaining the incurred loss in the approximation. This allows us to make another small change and, among all threshold-feasible items, always assign the *utility-maximizing* one. In turn, the routine becomes truthful, and we have to mainly care about showing the approximation.

Our strategy for setting the aforementioned thresholds can be interpreted as follows. Assume we take as a starting point not our prophet inequality for the bipartite case, but the one for general graphs with edge arrivals. Obviously, arrival orders for the bipartite one-sided model we consider here are highly restricted compared to general edge-arrival. However, assuming the graph is bipartite, each arriving edge consists of exactly one buyer b and one item i. Then, Algorithm 1 assigns prices p_b and p_i, where p_b corresponds to the aforementioned buyer's threshold we are planning to use.

Our mechanism works as follows: we determine greedy prices on the sample graph for all vertices just as described above. When a buyer b arrives, we present her not with the greedy price of each item, but instead charge for each item i the maximum over p_i and p_b. Then, we let the buyer choose what she likes best, i.e. we assign a utility-maximizing item.

Algorithm 5: Truthful Bipartite Prophet Matching

Set $M = \emptyset$
Compute the greedy matching M_S on the graph with edge weights according to sample S **for** *all*
 $e = \{b, i\} \in M_S$ **do**
  Set $p_b = p_i = s_e$
for *all vertices k not matched in M_S* **do**
  Set $p_k = 0$
for *each arriving buyer $b \in B$* **do**
  Set $F(b) = \{e = \{b, i\} \mid r_e \geq \max\{p_b, p_i\}$ and i is not matched in $M\}$
  Set $\tilde{e} = \{b, i^*\} = \arg\max\{r_e - \max\{p_b, p_i\} \mid e = \{b, i\} \in F(b)\}$
  Add $\tilde{e}$ to M with weight $r_{\tilde{e}}$
  Charge buyer b a price of $\max\{p_b, p_{i^*}\}$
return M

Here, M_S is again a greedy matching on the sample graph. As before (we omit this part of the analysis because it is analogous), one can imagine an offline version of Algorithm 5 that simply draws two values for each edge, and then goes through their set A in non-increasing order, deciding for each value considered whether it belongs to R or S (if this was not yet decided by the twin value for the same edge). We are able to prove the following result for Algorithm 5:

THEOREM 3.2. *For the problem of finding a maximum-weight matching in a bipartite graph $\mathcal{G}$ in the online vertex arrival model, Algorithm 5 is truthful and* 16-*competitive in expectation, i.e.,*

$$16 \cdot \mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\mathsf{OPT}\right],$$

where OPT *is the weight of an optimal matching in $\mathcal{G}$.*

Proof sketch. Truthfulness is immediate: on arrival, the unmatched items are offered to the buyer for prices determined independently of the buyer's values, and she is assigned one that is maximizing her utility. By misreporting, her utility can therefore only become worse.

The competitive analysis of the Algorithm 5 follows directly from the analysis of our algorithm for the matching problem on general graphs with edge-arrivals, Algorithm 1, since that analysis holds for any (adversarial) edge arrival order. See Appendix A for additional details.   □

3.3 Transversal Matroids We now show how our algorithm and analysis for the vertex-arrival model can be easily modified to yield an 8-competitive algorithm for the case of a *transversal matroid*. Recall that, given an undirected bipartite graph $\mathcal{G} = (B \cup I, E)$, a transversal matroid is defined over the ground set of buyers B, where a set $S \subseteq B$ is independent (i.e., feasible) if the vertices of S can be matched in $\mathcal{G}$ with a subset of the set of items I. Each buyer $b \in B$ is associated with a weight w_b drawn independently from distribution D_b, and *every* edge incident to buyer b inherits this weight.

We modify our algorithm for bipartite matching, Algorithm 3, to obtain an analogous SSPI for the setting of transversal matroids, Algorithm 6.

Algorithm 6: Transversal Matroid

Set $E^+ = \emptyset$, $M = \emptyset$

Choose an arbitrary ordering on the items of I // `For breaking ties within each buyer.`

Compute a greedy matching M_S on the graph with edge weights according to sample $S = \{s_b \mid b \in B\}$,
  breaking ties for each buyer according to the ordering on I

for *each* $e = \{b, i\} \in M_S$ **do**
    Set $p_b = p_i = s_b$

for *each vertex* k *not matched in* M_S **do**
    Set $p_k = 0$

for *each arriving buyer* $b \in B$ **do**
    Let i^* be the first item in the ordering on I such that $r_b \geq \max\{p_b, p_{i^*}\}$ and $\{b, i^*\} \in E$
    Add $\hat{e} = \{b, i^*\}$ to E^+ with weight $r_{\hat{e}}$ // `Only used for the analysis.`
    if i^* *is not matched in* M **then**
        Add $\hat{e}$ to M with weight $r_{\hat{e}}$

return M

In the vertex-arrival model for bipartite graphs, we assumed that the distribution of *all edges* is product form, even for the edges corresponding to a single buyer. We emphasize that it is not clear, in general, how to extend our techniques to allow for *arbitrary* correlation between edges of a single buyer. Indeed, since the definition of the set A in Algorithm 4 crucially assumes that the distribution of *all* edges is product form, and if the edge weights for each buyer were not independent, then the equivalence between the offline and online algorithms proved in Claim 3.1 no longer holds. However, our technique extends *directly* to a special type of correlation structure induced by the transversal matroid – the one where each buyer has the *same* weight for each item in its edge set.

By extending the arguments used to show Theorem 3.1, we can establish the following guarantee for this setting:

THEOREM 3.3. *For the problem of finding a maximum-weight independent set in a transversal matroid $\mathcal{M}$, Algorithm 6 is 8-competitive in expectation, i.e.,*

$$8 \cdot \mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\mathsf{OPT}\right],$$

where OPT *is the weight of a maximum-weight independent set in the transversal matroid $\mathcal{M}$.*

Proof sketch. The proof follows via careful modifications to the arguments used for Theorem 3.1: we construct a version of the offline algorithm where set A is a set of $2|B|$ *buyer* weights (as opposed to $2|E|$ *edge* weights in Algorithm 4). With this modification to the offline algorithm, the equivalence to Algorithm 6 follows analogously to before. The remaining proofs rely largely on the same probabilistic arguments as for the matching case, which primarily depend on the independence of edge weights for *different* buyers. We refer the reader to Appendix A for additional details. □

4 Budget-Additive Combinatorial Auctions

We now consider a special case of combinatorial auctions, where the buyers have budget-additive valuations. Specifically, each buyer $b \in B$ values every item $i \in I$ at some fixed value $v_b(i)$, drawn independently from a distribution for every item-buyer pair. However, once the total reward collected by a buyer b has reached a certain

budget C_b, additional items have no marginal contribution any more. More formally, we define $v_b(I')$ for any $I' \subseteq I$ as

$$v_b(I') = \min \left\{ \sum_{i \in I'} v_b(i), C_b \right\}.$$

We assume the budgets C_b to be fixed beforehand and known to the algorithm. Without loss of generality, we further assume that for all $i \in I$ and all $b \in B$, $v_b(i) \leq C_b$ (otherwise, we may pre-process those valuations exceeding C_b to have weight exactly C_b).

Similarly to Section 3, we assume that buyers arrive in an online adversarial order. Each time a buyer arrives, we observe the reward realizations of the buyer for all the items, and we can assign to her any collection of items that are not already assigned to a different buyer. Equipped with a single sample s_e for each buyer-item edge $e = \{b, i\}$, the goal is to maximize the expected reward collected against that of a prophet who knows all the reward realizations beforehand, and simply chooses the optimal assignment.

Our algorithm uses the following routine to compute a near-optimal solution G_S using the samples S. It starts by sorting in decreasing order the sample values $S = \{v_1, v_2, \ldots\}$, then, in that order, each edge is added to the solution G_S (i.e., an item is assigned to a buyer) whenever the item is available (i.e., has not been previously assigned) and its allocation does not exceed the according buyer's budget. When, for the first time, the algorithm tries to add an item to buyer b's bundle that would exceed her budget, then buyer b becomes *blocked* forever and cannot collect any more items (including the current one). We remark that, in the above greedy routine, items are only collected if their marginal contribution equals their value.

Once this greedy solution G_S is computed on the samples, our algorithm uses it to decide on the online allocation. In particular, we associate to each item i a threshold τ_i corresponding to the value of the buyer that gets the item in G_S, or to zero if i is unallocated. Every time a new buyer b arrives in the online phase, the items for which b's valuation beats the threshold, and which are still not allocated (in the actual solution M, not in G_S) are considered in decreasing order of b's weights. Each element is then actually allocated if it fits the buyer's budget in both M and in G_S. While it is clear what we mean by saying that the element fits in M, we need to specify what it means to say it also fits in G_S: when a buyer-item pair $e = \{b, i\}$ with value a is considered, which beats τ_i and fits into the budget of b in M, the algorithm computes $C_S(b, a)$, i.e. the total value of the items allocated to b in G_S restricted to *only* items with larger value than a. Then the item is actually allocated if and only if $C_S(b, a) + a \leq C_b$, and buyer b is not blocked in G_S after considering all samples larger than a. The formal pseudocode is given in Algorithm 7.

Algorithm 7: Budget-Additive Combinatorial Auctions

Set $E' = \emptyset$, $M = \emptyset$
Set $C_R(b) = 0$ for each buyer $b \in B$
Compute the greedy solution G_S with edge weights according to sample $S = \{s_e \mid e \in E\}$
Let $C_S(b, a)$ denote the total weight assigned to buyer b in G_S, after processing edges of sample-weight
  larger than a, or C_b in the case where buyer b is already blocked in G_S after considering all edges of
  sampled weight larger than a.
for *each $e = \{b, i\} \in G_S$* **do**
    Set $\tau_i = s_e$
for *each item i not assigned in G_S* **do**
    Set $\tau_i = 0$
for *each arriving buyer $b \in B$* **do**
    for *each edge $e = \{b, i\} \in E$ with weight r_e, in decreasing order of weight* **do**
        if $r_e > \tau_i$ *and* $r_e + C_S(b, r_e) \leq C_b$ **then**
            Add e to E' with weight r_e // `Only used for the analysis.`
            if $r_e + C_R(b) \leq C_b$ *and i has not yet been assigned in M* **then**
                Set $C_R(b) = C_R(b) + r_e$
                Add e to M with value r_e
return M

Equivalent offline algorithm. Following the usual proof technique, we state in Algorithm 8 an offline procedure that is easier to analyze and is equivalent to the online one.

Algorithm 8: Offline Simulation for Budget-Additive Combinatorial Auctions

Set $E' = \emptyset$, $G_S = \emptyset$, $M = \emptyset$, $I^S = \emptyset$
Set $C_S(b) = C_R(b) = 0$ for each $b \in B$
For each $e \in E$, draw from D_e two values $a_{e,1}$ and $a_{e,2}$
Order $A = \{a_{e,1}, a_{e,2} | e \in E\}$ in a decreasing fashion
for *each value $a \in A$ in the above order* **do**
    if *a corresponds to an edge $e = \{b, i\}$ that has never been observed before* **then**
        Flip a coin
        if *Heads* **then**
            Mark e as R-used
            if *$a + C_S(b) \leq C_b$ and $i \in I^S$* **then**
                Add e with value a to E'
        else
            Mark e as S-used
            if *$i \in I^S$* **then**
                if *$a + C_S(b) \leq C_b$* **then**
                    Set $C_S(b) = C_S(b) + a$
                    Add e to G_S with value a
                    Remove i from I^S
                else
                    Set $C_S(b) = C_b$
    else if *a corresponds to an edge $e = \{b, i\}$ which is R-used and $i \in I^S$* **then**
        if *$a + C_S(b) \leq C_b$* **then**
            Set $C_S(b) = C_S(b) + a$
            Add e to G_S with value a
            Remove i from I^S
        else
            Set $C_S(b) = C_b$
for *each $e = \{b, i\}$ in the same order as Algorithm 7* **do**
    if *$e \in E'$, $a + C_R(b) \leq C_b$, and i has not been assigned in M* **then**
        Add e to M with value a
        Set $C_R(b) = C_R(b) + a$
return M.

We start by noting that the two algorithms retain the same sets of interest. The proof of this claim follows using similar arguments as in Claims 2.1 and 3.1, and is given in Appendix A.

CLAIM 4.1. *The sets E', G_S, and M are distributed the same way when computed by Algorithm 7 as when computed by Algorithm 8.*

Correctness and competitive analysis. The correctness of Algorithm 7 follows by the fact that the algorithm never assigns a single item to more than one buyer.

We focus now on the analysis of the competitive guarantee of our policy. For any set of (weighted) edges $\hat{E}$, we denote by $\hat{E}(b)$ the (weighted) subset of edges in $\hat{E}$ that are incident to buyer b. As a first step, we establish that the greedy assignment G_S is a constant-approximation to the maximum-weight assignment:

LEMMA 4.1. *The solution G_S computed on the set of samples S is a 3-approximation to the optimal assignment over S.*

Proof. Let us denote by $\mathsf{OPT}(b)$ the items assigned to buyer b in an optimal assignment, and $v_b^*(i)$ the value buyer

b realizes from item i in OPT. Note that every buyer might (w.l.o.g.) have at most one fractional item i_b in his OPT-bundle, where, in that case, $v_b^*(i_b)$ denotes only a fraction of $v_b(i_b)$, accordingly.

We compute our greedy solution G_S in a way that might miss out on some of the value realized in OPT. Let's focus on a buyer b and OPT(b). For any $i \in$ OPT(b), one of the following cases is true:

1. i is also part of b's greedy bundle $G_S(b)$.

2. i is part of some other buyer's $G_S(b')$, and $v_{b'}(i) \geq v_b^*(i)$.

3. i is assigned for a lower value than in OPT, or not assigned at all.

Let us call items for which these cases hold I_1, I_2 and I_3, accordingly. The value OPT(I_1) realized on items of the first case is clearly at most that realized by $G_S(I_1)$, since the marginal contribution of each item in G_S always equal to its value. Further, for the same reason as above, together with the fact that each item in I_2 gets a better value in G_S, it follows that OPT(I_2) $\leq G_S(I_2)$. Finally, in order to conclude the proof, it suffices to bound the loss incurred due to the items in I_3. If $i \in I_3$, $i \in$ OPT(b), then at the point where value $v_b(i)$ is considered by the greedy, either (i) item i is no longer available, or (ii) buyer b is blocked, or (iii) adding $v_b(i)$ would exceed C_b. Clearly, case (i) cannot be true, since if i had been already assigned by G_S, then because of the decreasing order where the greedy algorithm parses the edge, item i would have been in I_2. Therefore, it has to be that either b's budget was running full, or b was blocked. In both cases, and again due to the decreasing order of consideration, $G_S(b) \geq \frac{C_b}{2}$.

Now, let us write the value realized by OPT on item set $I' \subseteq I$ as OPT(I'). According to the previous considerations, we get

$$w(G_S) \geq \text{OPT}(I_1) + \text{OPT}(I_2), \text{ and } w(G_S) \geq \frac{1}{2} \cdot \text{OPT}(I_3)$$

because buyers with *any* I_3-items in OPT(b) have at least half-full budget. Together, this yields

$$G_S \geq \max\left\{(\text{OPT}(I_1) + \text{OPT}(I_2)), \frac{1}{2}\text{OPT}(I_3)\right\} \geq \min_{g,f \geq 0: g+f=1} \max\left\{g, \frac{f}{2}\right\} \cdot \text{OPT} = \frac{1}{3}\text{OPT}.$$

$\square$

In the above algorithm, E' is the (weighted) set of all edges with a reward greater than the according item threshold, which are also considered before the according buyer is blocked in G_S. Note that similarly to previous proofs, and due to the fact that conditions for adding an element to E' or G_S are exactly the same, $\mathbb{E}[w(E')] \geq \frac{1}{2} \cdot \mathbb{E}[w(G_S)]$: this is true because every edge considered for the *first* time lands in both sets equally likely, and an edge considered for the *second* time can only be added to G_S, but then with at most the same value as its first occurrence.

As in the previous sections, the proof of our claim relies on constructing a special subset of the edges in E', called the *safe edges*:

DEFINITION 4.1. *We call an edge $e = \{b, i\} \in E$ safe for buyer b if the following conditions are true:*

1. *Edge e is in E' with some weight r_e.*

2. *There is no edge in E' that is incident to item i and has weight smaller than r_e.*

However, opposed to our earlier proofs, being safe in the sense of belonging to $E_{\text{safe}} = \{e \in E' \mid \exists b \in B \text{ s.t. } e \text{ is safe for } b\}$ does not suffice for us to recover an edge in the actual SSPI: still, it might happen that at the point where buyer b arrives, edge $e = \{b, i\}$ cannot be (fully) realized because C_b is exceeded. Indeed, we will only be able to extract the value from E_{safe} that *does not* exceed the capacity C_b of each buyer b. To bound this, we will have to intersect E_{safe} with a (weighted) set E^+, which incorporates the capacity constraints for each buyer b:

$$E^+(b) = \left\{a \in E'(b) \mid a + \sum_{a' \in E'(b),\, a' > a} a' \leq C_b\right\}.$$

We now show a crucial property of E^+ – namely, that its expected weight can be lower-bounded using that of the greedy solution, G_S:

LEMMA 4.2. *Let G_S be the greedy matching on the set of samples. Then,*

$$\mathbb{E}\left[w(E^+)\right] \geq \frac{1}{4} \cdot \mathbb{E}\left[w(G_S)\right].$$

Proof. For any fixed value realization A and any buyer b, we define

$$X(b) = \{(e, a) \in G_S(b) \cup E'(b) \mid a \text{ is the larger value drawn for } e \text{ in } A\}.$$

where the tuple (e, a) refers to adding edge e with a weight $w(e) = a$. Informally, for each buyer b, the set $X(b)$ contains all the values $a_{e,1}$ (i.e., the larger realization) for each edge $e \in G_S(b) \cup E'(b)$.

Note that deterministically, $w(X) \geq w(G_S)$, since for every edge $e \in G_S$ that is weighted with the smaller realization $a_{e,2}$, it has to be that $a_{e,1}$ belongs to E' and, thus, also belongs to X. Further, we define

$$X^+(b) = \left\{(e, a) \in X(b) \mid a + \sum_{(e,a') \in X(b),\, a' > a} a' \leq C_b\right\}.$$

as the subset of weighted edges $(e, a) \in X(b)$ such that a, together with all weights $a' > a$ in $X(b)$, does not exceed the capacity C_b of buyer b.

Given the above definitions, our proof consists of showing that, for any buyer $b \in B$, it holds
(i) $\mathbb{E}\left[w(X^+(b))\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[w(G_S(b))\right]$ and (ii) $\mathbb{E}\left[w(E^+(b))\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[w(X^+(b))\right]$.

In order to show (i), we first rewrite the expected weight of $X^+(b)$, for any buyer b, as

$$\mathbb{E}\left[w(X^+(b))\right] = \Pr\left[w(X(b)) > C_b\right] \cdot \mathbb{E}\left[w(X^+(b)) \mid w(X(b)) > C_b\right]$$
$$+ \Pr\left[w(X(b)) \leq C_b\right] \cdot \mathbb{E}\left[w(X^+(b)) \mid w(X(b)) \leq C_b\right].$$

Now if at some point, an edge with value a was added to $X(b)$ that made its weight exceed C_b, at this point in time it already was true that $w(X^+(b)) \geq \frac{C_b}{2}$, due to decreasing order. Thus, it follows that

$$\mathbb{E}\left[w(X^+(b))\right] \geq \Pr\left[w(X(b)) > C_b\right] \cdot \frac{C_b}{2} + \Pr\left[w(X(b)) \leq C_b\right] \cdot \mathbb{E}\left[w(X^+(b)) \mid w(X(b)) \leq C_b\right].$$

Recall that, independently of the realization of A and the random coin flips, it holds $w(G_S(b)) \leq w(X(b))$. Thus, since the sets $X(b)$ and $X^+(b)$ are identical in the case that $X(b) \leq C_b$, it follows that

$$\mathbb{E}\left[w(X^+(b)) \mid w(X(b)) \leq C_b\right] = \mathbb{E}\left[w(X(b)) \mid w(X(b)) \leq C_b\right] \geq \mathbb{E}\left[w(G_S(b)) \mid w(X(b)) \leq C_b\right].$$

In addition, it holds (since our greedy routine ignores partial items) $w(G_S(b)) \leq C_b$, thus also $\mathbb{E}\left[w(G_S(b)) \mid w(X(b)) > C_b\right] \leq C_b$, and therefore

$$\mathbb{E}\left[w(X^+(b))\right] \geq \Pr\left[w(X(b)) > C_b\right] \cdot \frac{1}{2} \cdot \mathbb{E}\left[w(G_S(b)) \mid w(X(b)) > C_b\right]$$
$$+ \Pr\left[w(X(b)) \leq C_b\right] \cdot \mathbb{E}\left[w(X^+(b)) \mid w(X(b)) \leq C_b\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[w(G_S(b))\right].$$

Now in order to prove statement (ii), we fix some realization of A, some value $a \in A$, and some run of the offline algorithm up to the point where a is considered. Note that if a is the weight of the second occurrence of the according edge, it will not be considered for $X(b)$ or $E'(b)$. However if it is the first and price-feasible, it is added to $X(b)$ with probability 1 (as it is added to either G_S or E'), and to $E'(b)$ with probability $\frac{1}{2}$. Now, if $(e, a) \in X(b) \cap X^+(b)$, then still, it is added also to $E'(b)$ and therefore, $E^+(b)$, with probability $\frac{1}{2}$. Thus, it follows that $\mathbb{E}\left[w(E^+(b))\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[w(X^+(b))\right]$.

The proof of the Lemma follows by combining inequalities (i) and (ii). $\square$

Next, we lower bound the probability that an edge $e = \{i, b\} \in E^+$ is safe for buyer b:

LEMMA 4.3. *For any buyer $b \in B$ and edge $e = \{b, i\}$ which is in E^+ with non-zero probability, we have that*

$$\Pr\left[e \text{ is safe for buyer } b \mid e = \{b, i\} \in E^+\right] \geq \frac{1}{2}.$$

Proof. Consider the time when, during a run of Algorithm 8, e is added to E^+. By construction, the algorithm only adds edges which are R-used and are being processed for the first time.

Now, consider the next time when an edge $e' = \{b', i\}$ is being considered by the algorithm with weight a which allows it to be allocated in G_S. Now, if e' is being processed for the second time, then it must be R-used and added to G_S (and thus, at this point, i is removed from I^S). Thus, in this case, no other edge $e'' = \{b'', i\}$ with value smaller than r_e could be added to E^+, guaranteeing that both conditions of Definition 4.1 are satisfied. In the case when e' is processed for the first time, then, with probability $1/2$, e is S-used and added to G_S, again guaranteeing that no other element of smaller weight is added to E^+, and thus that the conditions of Definition 4.1 are satisfied. $\square$

LEMMA 4.4. *Recall that E_{safe} is the set of edges $e = \{b, i\} \in E'$ which are safe for some buyer b. Then,*

$$\mathbb{E}\left[w(E_{safe} \cap E^+)\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[w(E^+)\right].$$

Proof. Here, the argument is essentially the same as in Lemma 2.2, replacing the use of Lemma 2.1 with Lemma 4.3. In particular, note that, taking (without loss of generality) $a_{e,1}$ to be the larger realization of the edge weight for e,

$$\mathbb{E}\left[w(E_{\text{safe}} \cap E^+)\right] = \mathbb{E}\left[\sum_{e=\{b,i\} \in E^+} r_e \cdot \mathbb{1}_{\{e \text{ is safe for } b\}}\right] = \sum_{e=\{b,i\} \in E} a_{e,1} \cdot \Pr\left[e \text{ is safe for } b, e \in E^+\right].$$

Thus, by Lemma 4.3,

$$\Pr\left[e \text{ is safe for } b, e \in E^+\right] = \Pr\left[e \text{ is safe for } b \mid e \in E^+\right] \Pr\left[e \in E^+\right] \geq \frac{1}{2} \cdot \Pr\left[e \in E^+\right].$$

By combining these two results, we conclude that

$$\mathbb{E}\left[w(E_{\text{safe}} \cap E^+)\right] \geq \frac{1}{2} \sum_{e=\{b,i\} \in E} a_{e,1} \cdot \Pr\left[e \in E^+\right] = \mathbb{E}\left[w(E^+)\right],$$

as desired. $\square$

LEMMA 4.5. *Let M be the assignment obtained in a run of Algorithm 8. Then,*

$$\mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[w(E_{safe} \cap E^+)\right].$$

Proof. We prove a stronger result – namely, for *any* run of Algorithm 8, $w(M) \geq w(E_{\text{safe}} \cap E^+)$. Consider any edge $e = \{b, i\} \in E^+$ which is safe for buyer b. By construction of the set E^+, and since edges in E' are processed in decreasing order of weight, when e is processed by the algorithm, it must be the case that either e is added to M, or some other edge $e' = \{b', i\} \in E'$ was already chosen incident to item i before edge e was processed during the online phase (since the algorithm only accepts items in the set E'). We emphasize here that by definition and due to the weight-decreasing order, an edge $e = \{b, i\} \in E^+$ can *never* be rejected because it exceeds the buyer's capacity C_b. Since e is safe for buyer b, and since Algorithm 8 *never* collects truncated rewards by construction , it must be the case that $r_{e'} > r_e$. Therefore, e' cannot be safe for buyer b', since the second condition Definition 4.1 is violated by edge e. Hence, for any run of Algorithm 8 for which e is safe for buyer b, an element of weight at least r_e is collected in M. By aggregating this result for all edges in E^+, we reach the desired result. $\square$

THEOREM 4.1. *For the problem of finding a maximum-weight assignment in a budget-additive combinatorial auction in the online vertex arrival model, Algorithm 7 is 24-competitive in expectation, i.e.,*

$$24 \cdot \mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\mathsf{OPT}\right],$$

where OPT is the weight of a maximum-weight assignment.

Proof. Combining the results of Lemma 4.5 and Lemma 4.4, we can relate the expected reward collected by Algorithm 7 with that collected by E^+:

$$\mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[w(E_{\text{safe}} \cap E^+)\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[w(E^+)\right].$$

Then, combining the result of Lemma 4.2, which allows us to relate the expected weights of E^+ and the greedy solution with respect to the samples, G_S, with Lemma 4.1, which relates the expected weights of the greedy and optimal solutions, we have that

$$\mathbb{E}\left[w(E^+)\right] \geq \frac{1}{4} \cdot \mathbb{E}\left[w(G_S)\right] \geq \frac{1}{12} \cdot \mathbb{E}\left[\mathsf{OPT}\right].$$

Combining these results, we conclude that

$$\mathbb{E}\left[w(M)\right] \geq \frac{1}{24} \cdot \mathbb{E}\left[\mathsf{OPT}\right],$$

as claimed. □

5 From α-Partition to Single-Sample Prophet Inequalities

In this section, we turn our attention to the SSPI problem under matroid feasibility constraints. We show how the recent work of Rubinstein et al. [2020] can be applied on an interesting class of matroids that satisfy a certain property, called α-partition. For this class, we provide improved competitive guarantees (essentially by a factor of 2) for the SSPI problem, comparing to those following from the reduction to OOS, presented in Azar et al. [2014].

We refer to a matroid $\mathcal{M} = (E, \mathcal{I})$ over a ground set E as a *simple* partition matroid[5], if there exists some partition $\bigcup_{l \in [k]} E_l$ of E such that $I \in \mathcal{I}$ if and only if $|I \cap E_l| \leq 1$ for each $l \in [k]$. In other words, a set is independent only if it contains at most one element from each set of the partition.

We consider the following property of several matroids, called α-partition:

DEFINITION 5.1. (α-PARTITION PROPERTY (SLIGHTLY ADAPTED FROM [BABAIOFF ET AL., 2009])) *A matroid $\mathcal{M} = (E, \mathcal{I})$ satisfies an α-partition property for some $\alpha \geq 1$ if for any weight vector w on the elements, after sequentially observing the weight of a (possibly random) subset $S \subset E$ of the elements selected independently of the weights, one can define a simple partition matroid $\mathcal{M}' = (E', \mathcal{I}')$ on a ground set $E' \subseteq E \setminus S$, such that*

$$\mathbb{E}\left[\max_{I' \in \mathcal{I}'} w(I')\right] \geq \frac{1}{\alpha} \cdot \max_{I \in \mathcal{I}} w(I) \qquad \text{and} \qquad \mathcal{I}' \subseteq \mathcal{I},$$

where the expectation is taken over any randomness in the partitioning procedure.

In fact the above definition is a slight adaptation of one in [Babaioff et al., 2009]. Specifically, we additionally capture the case where the transformation may observe the weight of a subset $S \subset E$ of the elements, and that this S is never included in the ground set of the produced partition matroid. Notice that the above definition also permits transformations that do not depend on the weights of *any* sample (i.e., $S = \emptyset$), or that are deterministic.

We prove the following meta-theorem for any matroid with an α-partition property:

THEOREM 5.1. *For any matroid $\mathcal{M}$ that satisfies an α-partition property for some $\alpha \geq 1$, there exists a 2α-competitive policy for the corresponding SSPI problem. Further, if the α-partitioning can be performed in polynomial time, then the policy is also efficient.*

Proof. Given any matroid $\mathcal{M} = (E, \mathcal{I})$ which satisfies an α-partition property, we describe how to construct a simple 2α-competitive policy for the SSPI on $\mathcal{M}$. The policy proceeds in two phases: (Offline phase) We construct a simple partition matroid $\mathcal{M}' = (E', \mathcal{I}')$ from $\mathcal{M}$, using its α-partition property. For each element $e \in S \subset E$ that needs to be observed by the transformation, we feed the corresponding sample s_e. Let $\bigcup_{l \in [k]} E_l = E'$ be the

---

[5]We use the term "simple" in order to distinguish the partition matroid from its common definition, where, from each set E_l of the partition, more than one element might be collected.

constructed partition. For each group E_l of $\mathcal{M}'$, we set a threshold $\tau_l = \max_{e \in E_l} s_e$, equal to the value of the largest sample of the elements included in E_l. Finally, we initialize $I = \emptyset$. (Online phase) For each element $e \in E$ arriving in adversarial order, we immediately reject it (without even observing the associated reward), if $e \notin E'$. Otherwise, assuming that $e \in E_l$ for some $l \in [k]$, we accept e and add it to I if and only if (i) $I \cap E_l = \emptyset$ (i.e., no other element in E_l has been accepted so far) and (ii) the reward satisfies $r_e > \tau_l$.

We now show that the above policy is 2α-competitive for the SSPI problem on $\mathcal{M}$. First observe that, by construction of the policy, the set of elements collected satisfies $I \in \mathcal{I}'$ and, thus, $I \in \mathcal{I}$ by definition of the α-partition property. In order to establish the competitiveness of the policy, we first note that the maximum-reward independent set of $\mathcal{M}'$ is simply the collection of maximum-reward elements in each group of the partition. Notice further that the online phase of our policy consists of running several instances of the algorithm of Rubinstein et al. [2020], one for each partition. Therefore, by linearity of expectation, our policy collects in expectation a $1/2$-fraction of the expected maximum-reward within each partition and, hence, is 2-competitive against the optimal policy for M'. Finally, given that the partitioning is performed using the samples, the expected maximum-reward independent set of $\mathcal{M}'$ has the same expectation as if the partitioning was performed using the rewards, since the samples and rewards are identically distributed. By applying Definition 5.1, we can conclude that our policy is 2α-competitive for $\mathcal{M}$. □

REMARK 5.1. *We remark that Theorem 5.1 can be extended in the case of other combinatorial problems (not necessarily matroids) that satisfy a partitioning property, similar to Definition 5.1.*

As an application of Theorem 5.1, we obtain improved SSPIs for a number of settings considered in Azar et al. [2014], displayed in last four entries in Table 1. Indeed, the improvement in these results comes by noticing that each of these matroid environments satisfy the α-partition property for constant α, and rely on a 4-competitive single-choice OOS. As a result of Theorem 5.1 and Rubinstein et al. [2020], we can replace this OOS with a 2-competitive SSPI, thus improving the competitive guarantee by a factor of 2.

6 From Order-Oblivious Secretaries to Pointwise-SSPIs

Since *direct* SSPIs not only appear to offer better approximation than OOS policies, but also obviously have access to more information (the samples) comparing to an OOS algorithm, one can hope to derive constant-factor SSPIs even where according OOS policies fail to exist. For example, a major open problem in optimal stopping theory is the existence of $\mathcal{O}(1)$-competitive policies for the matroid secretary problem. While Babaioff, Immorlica, and Kleinberg [Babaioff et al., 2007] conjectured the existence of such a policy, and $\mathcal{O}(1)$-competitve policies are known for many special cases (see, e.g., Soto et al. [2021]), the state-of-the-art for general matroids is a $\mathcal{O}(\log\log(\text{rank}))$-competitive policy due to Feldman et al. [2014], Lachish [2014]. In stark contrast, Kleinberg and Weinberg [2012] give an *optimal* 2-competitive policy for the related matroid prophet inequality problem under full distributional knowledge. Thus, given that the existence of an $\mathcal{O}(1)$-competitive policy for the SSPI problem on general matroids is a persistent open question, it is natural to ask whether the problem exhibits similarities in terms of hardness with its (order-oblivious) secretary counterpart.

In this section, we make progress towards the above question, by relating the OOS problem with a wide class of SSPI policies, which we refer to as "pointwise"-SSPIs (or "P-SSPIs").

DEFINITION 6.1. (P-SSPI) *An SSPI policy is called P-SSPI, if it maintains its competitive guarantee when the reward and sample of each element e are generated as follows: given two arbitrary non-negative weights $a_{e,1}$ and $a_{e,2}$, we flip an independent fair coin to decide whether $r_e = a_{e,1}$ and $s_e = a_{e,2}$, or the opposite.*

Notice that, as opposed to the SSPI problem, where each reward and sample are drawn independently from the same distribution, P-SSPI allows for the reward and sample to be *correlated*. Further, by the exact same reduction (and proof) presented in [Azar et al., 2014], which reduces SSPI to OOS, one can in fact show the following stronger result:

THEOREM 6.1. ([AZAR ET AL., 2014]) *Given any α-competitive OOS, one can construct an α-competitive P-SSPI.*

An important observation is that every known SSPI policy is *also* a P-SSPI policy with the same competitive guarantee.

OBSERVATION 6.1. *Every known SSPI policy, including the single-choice SSPI of Rubinstein et al. [2020], any SSPI in [Azar et al., 2014], and our policies, falls into the class of P-SSPIs.*

Perhaps surprisingly, we can establish the following result, which is a partial converse to the reduction in Lemma 6.1:

THEOREM 6.2. *For any downward-closed feasible set of elements, given any α-competitive P-SSPI, one can construct a 2α-competitive OOS.*

Proof. Given an α-competitive policy P for the P-SSPI problem, we construct a policy A for the OOS problem as follows: (Phase 1) Let $k \sim \text{Binomial}(n, 1/2)$ be a random number of elements. Observe (without collecting) the rewards of the first k elements in uniformly random order, and feed them to P as the *samples* of the corresponding elements. For the elements that are not yet observed, we feed to P *zeros* as the corresponding samples. (Phase 2) For the remaining elements arriving in adversarial order, the OOS policy A mimics the decisions of P (i.e., the OOS policy accepts if and only if P accepts), skipping the elements that are parsed in Phase 1.

We now show that the above policy is 2α-competitive for the OOS problem. Let $\text{OPT}(\text{OOS})$ be the optimal reward for the OOS problem on some instance, and let $\text{OPT}(\text{P-SSPI})$ be the prophet's reward on the corresponding instance where the rewards of k elements chosen uniformly at random are set to zero. For our selection of k, it is not difficult to verify that the reward of each element is zero (and its weight is provided to P as a sample) with probability half. Hence, given that the optimal solution to OOS is feasible for P-SSPI and the reward of each element "survives" with probability half, it follows that

$$\mathbb{E}\left[\text{OPT}(\text{P-SSPI})\right] \geq \frac{1}{2} \cdot \text{OPT}(\text{OOS}).$$

Let $w(P)$ and $w(A)$ be the reward collected by policy P and A, respectively, in the above reduction. Notice that by construction of our reduction, every element that is parsed in Phase 1 of A, has zero reward and, thus, is never collected by P in the online phase, without loss of generality. Given that these elements are skipped by A in Phase 2 and, for the rest of the elements, A mimics the decisions of P, we can see that $w(P) = w(A)$. Finally, since P is an α-competitive P-SSPI, we can conclude that

$$\mathbb{E}\left[w(A)\right] = \mathbb{E}\left[w(P)\right] \geq \frac{1}{\alpha} \cdot \mathbb{E}\left[\text{OPT}(\text{P-SSPI})\right] \geq \frac{1}{2\alpha} \cdot \text{OPT}(\text{OOS}),$$

thus showing that A is a 2α-competitive OOS policy. We remark that the downward-closedness assumption on the feasible set is only used for guaranteeing that rejecting elements in Phase 1 of A cannot lead to an infeasible solution. $\square$

An immediate corollary of Observation 6.1 and Theorem 6.2 is that any *known* SSPI policy effectively provides an algorithm for the corresponding (order-oblivious) secretary problem.

Note that by applying Theorem 6.2 to Theorem 4.1, we obtain a 48-competitive OOS policy for budget-additive combinatorial auctions. To the best of our knowledge, this is the first order-oblivious secretary policy for this problem.

By Theorem 6.2, solving the P-SSPI problem on a downward-closed feasible set (including general matroids) *immediately implies* an OOS policy on the same feasible set. Hence, unless there exists a $\mathcal{O}(1)$-competitive SSPI which is *not* a P-SSPI, the SSPI problem is at least as *as hard* as the OOS problem (up to constant factors)! Since all currently known SSPI are P-SSPI, it is natural to ask whether this observation can be generalized:

Given any α-competitive SSPI policy, does there also exist an $\mathcal{O}(\alpha)$-competitive P-SSPI policy? Or, stated differently, are the classes of SSPI and P-SSPI equivalent up to constant factors?

In case of a positive answer to the previous problem, solving the constant-factor SSPI problem on general matroids would imply also resolving the matroid secretary conjecture of Babaioff et al. [2007]. In any case, and more generally: since P-SSPIs provide constant-factor approximations to the same class of problems as OOSs do, only *non*-P-SSPI might be more powerful, i.e., only with such policies, the SSPI paradigm can ever cover a greater scope of settings than OOS.

Acknowledgement

References

S. Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM J. Comput.*, 43(2):930–972, 2014.

S. Alaei, M. Hajiaghayi, and V. Liaghat. Online prophet-inequality matching with applications to ad allocation. In *EC'12*, pages 18–35, 2012.

P. Azar, S. Micali, C. Daskalakis, and S. M. Weinberg. Optimal and efficient parametric auctions. In *SODA'13*, pages 596–604, 2013.

P. D. Azar, R. Kleinberg, and S. M. Weinberg. Prophet inequalities with limited information. In *SODA'14*, pages 1358–1377, 2014.

M. Babaioff, N. Immorlica, and R. Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA'07*, pages 434–443, 2007.

M. Babaioff, M. Dinitz, A. Gupta, N. Immorlica, and K. Talwar. Secretary problems: weights and discounts. In *SODA'09*, pages 1245–1254, 2009.

S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan. Multi-parameter mechanism design and sequential posted pricing. In *STOC'10*, pages 311–320, 2010.

E. H. Clarke. Multipart pricing of public goods. *Public choice*, 11:17–33, 1971.

J. Correa, P. Dütting, F. Fischer, and K. Schewior. Prophet inequalities for iid random variables from an unknown distribution. In *EC'19*, pages 3–17, 2019.

J. R. Correa, A. Cristi, B. Epstein, and J. A. Soto. The two-sided game of googol and sample-based prophet inequalities. In *SODA'20*, pages 2066–2081, 2020a.

J. R. Correa, A. Cristi, B. Epstein, and J. A. Soto. Sample-driven optimal stopping: From the secretary problem to the i.i.d. prophet inequality. *CoRR*, abs/2011.06516, 2020b.

J. R. Correa, P. Dütting, F. A. Fischer, K. Schewior, and B. Ziliotto. Unknown I.I.D. prophets: Better bounds, streaming algorithms, and a new impossibility. In *ITCS'21*, pages 86:1–86:1, 2021.

P. Dhangwatnotai, T. Roughgarden, and Q. Yan. Revenue maximization with a single sample. In D. C. Parkes, C. Dellarocas, and M. Tennenholtz, editors, *EC'10*, pages 129–138, 2010.

N. B. Dimitrov and C. G. Plaxton. Competitive weighted matching in transversal matroids. *Algorithmica*, 62(1-2): 333–348, 2012.

P. Dütting and R. Kleinberg. Polymatroid prophet inequalities. In *ESA'15*, pages 437–449, 2015.

P. Dütting, M. Feldman, T. Kesselheim, and B. Lucier. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM J. Comput.*, 49(3):540–582, 2020a.

P. Dütting, T. Kesselheim, and B. Lucier. An O(log log m) prophet inequality for subadditive combinatorial auctions. In *FOCS'20*, pages 306–317, 2020b.

P. Dütting, F. Fusco, P. Lazos, S. Leonardi, and R. Reiffenhäuser. Efficient two-sided markets with limited information. In *STOC'21*, pages 1452–1465, 2021.

S. Ehsani, M. Hajiaghayi, T. Kesselheim, and S. Singla. Prophet secretary for combinatorial auctions and matroids. In *SODA'18*, pages 700–714, 2018.

T. Ezra, M. Feldman, N. Gravin, and Z. G. Tang. Online stochastic max-weight matching: Prophet inequality for vertex and edge arrival models. In *EC'20*, pages 769–787, 2020.

M. Feldman, O. Svensson, and R. Zenklusen. A simple O (log log (rank))-competitive algorithm for the matroid secretary problem. In *SODA'14*, pages 1189–1201, 2014.

M. Feldman, N. Gravin, and B. Lucier. Combinatorial auctions via posted prices. In *SODA'15*, pages 123–135, 2015.

M. Feldman, O. Svensson, and R. Zenklusen. Online contention resolution schemes. In *SODA'16*, pages 1014–1033, 2016.

N. Gravin and H. Wang. Prophet inequality for bipartite matching: Merits of being simple and non adaptive. In *EC'19*, pages 93–109, 2019.

T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

M. T. Hajiaghayi, R. Kleinberg, and T. Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI'07*, pages 58–65, 2007.

H. Kaplan, D. Naori, and D. Raz. Competitive analysis with a sample and the secretary problem. In *SODA'20*, pages 2082–2095, 2020.

H. Kaplan, D. Naori, and D. Raz. Online weighted bipartite matching with a sample. *CoRR*, abs/2104.05771, 2021.

R. Kleinberg and S. M. Weinberg. Matroid prophet inequalities. In *STOC'12*, pages 123–136, 2012.

N. Korula and M. Pál. Algorithms for secretary problems on graphs and hypergraphs. In *ICALP'09*, pages 508–520, 2009.

U. Krengel and L. Sucheston. Semiamarts and finite values. *Bull. Am. Math. Soc.*, 83:745–747, 1977.

U. Krengel and L. Sucheston. On semiamarts, amarts, and processes with finite value. *Adv. in Probab. and Related Topics*, 4:197—266, 1978.

O. Lachish. O(log log rank) competitive ratio for the matroid secretary problem. In *FOCS'14*, pages 326–335, 2014.

T. Ma, B. Tang, and Y. Wang. The simulated greedy algorithm for several submodular matroid secretary problems. *Theory Comput. Syst.*, 58(4):681–706, 2016.

R. B. Myerson. Optimal auction design. *Math. Oper. Res.*, 6(1):58–73, 1981.

A. Rubinstein. Beyond matroids: secretary problem and prophet inequality with general constraints. In *STOC'16*, pages 324–332, 2016.

A. Rubinstein and S. Singla. Combinatorial prophet inequalities. In *SODA'17*, pages 1671–1687, 2017.

A. Rubinstein, J. Z. Wang, and S. M. Weinberg. Optimal single-choice prophet inequalities from samples. In *ITCS'20*, pages 60:1–60:10, 2020.

E. Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *Ann. Probab.*, 12:1213–1216, 1984.

J. A. Soto. Matroid secretary problem in the random-assignment model. *SIAM J. Comput.*, 42(1):178–211, 2013.

J. A. Soto, A. Turkieltaub, and V. Verdugo. Strong algorithms for the ordinal matroid secretary problem. *Math. Oper. Res.*, 46:642–673, 2021.

W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *J. Finance*, 16(1):8–37, 1961.

A Appendix

CLAIM 3.1. *The sets E^+, M_S and M are distributed the same way when computed by Algorithm 3 as when computed by Algorithm 4.*

Proof. The proof is analogous to the one of Claim 2.1. The sets R and S in the offline and online setting follow the same distribution as argued in the previous Section, and M_S is still the greedy matching with respect to the samples S, in both Algorithm 3 and Algorithm 4. We know that M is extracted in the same way by E^+ by the two algorithms, so we just need to argue that for each fixed realization of the edge weights, the online and offline versions of the edge sets E^+ coincide.

In Algorithm 3, E^+ contains *at most* one edge $e = \{b, i\}$ for each buyer $b \in B$. This is the edge of *largest* reward that is above both prices p_b and p_i, if such an edge exists. Now, in Algorithm 4, observe that, once an edge $e = \{b, i\}$ is added to E^+, the buyer b is removed from the set B^R. Therefore, in this algorithm also, at most one edge can be added to E^+ for each buyer. Now, observe that if the edge is added to E^+, then it must be over the price of b and i. Indeed, this follows because, in order for e to be added to E^+, it must be the case that $b \in B^R$ (and, by construction, also in B^S) and $i \in I^S$, which implies that b and i have not yet been matched in the greedy sample solution, and thus, because of the greedy traversal order, its reward must be above both prices. Finally, we observe that, again due to the greedy traversal order, if e is added to E^+, then it must be the *largest*-reward edge which is above the prices p_b and p_i. Hence E^+ follows the same distribution in both algorithms. □

THEOREM A.1. *For the problem of finding a maximum-weight matching in a bipartite graph $\mathcal{G}$ in the online vertex arrival model, Algorithm 5 is truthful and 16-competitive in expectation, i.e.,*

$$16 \cdot \mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\mathsf{OPT}\right],$$

where OPT is the weight of an optimal matching in $\mathcal{G}$.

Proof. Truthfulness is immediate: on arrival, the unmatched items are offered to the buyer for prices determined independently of the buyer's values, and she is assigned one that is maximizing her utility. By misreporting, her utility can therefore only become worse.

The competitive analysis of the Algorithm 5 follows closely that of our algorithm for the matching problem on general graphs with edge-arrivals, Algorithm 1. Let E' be the set of *price-feasible* edges, namely, E' contains any edge $e = \{b, i\}$ such that $r_e \geq \max\{p_b, p_i\}$. Notice that, given both in Algorithm 1 and in Algorithm 5 the prices are computed in the exact same way (via the greedy matching solution using the S), E' has the exactly the same meaning in both algorithms. Further, note that $\bigcup_{i \in I} F(i) \subseteq E'$, and that the arrival order, be it edge-arrival or vertex-arrival, has no influence whatsoever on the computation of set E'.

Therefore, we can establish, in exactly the same way as in the proof of Theorem 2.1, that

$$\mathbb{E}\left[\sum_{b \in B} \mathrm{val}_{E'}(b)\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[\sum_{b \in B} \mathrm{val}_{M_S}(b)\right] = \frac{1}{2} \cdot \mathbb{E}\left[w(M_S)\right] \geq \frac{1}{4} \cdot \mathbb{E}\left[\mathsf{OPT}\right],$$

where by $\mathrm{val}_{\hat{E}}(b)$ we denote the maximum-weight over all edges of a weighted set $\hat{E} \subseteq E$ adjacent to buyer b.

Let e_b be the maximum-reward edge in E' adjacent to buyer b, if such an edge exists. By applying Definition 2.1 for the edge-arrival case, we say that an edge $e = \{b, i\}$ is *safe for buyer b*, if e is the only edge in E' adjacent to b, and there is no edge in E' that is adjacent to item i and has smaller weight than r_e. By the exact same arguments as in Lemma 2.1, for any buyer b and incident edge e such that the probability of $e_b = e$ is non-zero, we have that $\mathbb{P}[e_b$ is safe for $b \mid e_b = e] \geq \frac{1}{4}$. Thus, for any realization of the edge weights in A, by the analysis of Lemma 2.2, we have that

$$\mathbb{E}\left[\sum_{b \in B} r_{e_b} \mathbb{1}_{\{e_b \text{ is safe for } b\}}\right] \geq \frac{1}{4} \cdot \mathbb{E}\left[\sum_{b \in B} r_{e_b}\right] = \frac{1}{4} \cdot \mathbb{E}\left[\sum_{b \in B} \mathrm{val}_{E'}(b)\right] \geq \frac{1}{16} \cdot \mathbb{E}\left[\mathsf{OPT}\right].$$

Again, since our proof for the edge-arrival case holds for any (adversarial) arrival order, it can be easily verified that Algorithm 5 is going to collect a reward of at least the weight of the safe edges in E' in expectation. Indeed,

by following exactly the arguments of Lemma 2.3, we can show that

$$\mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\sum_{b \in B} r_{e_b} \mathbb{1}_{\{e_b \text{ is safe for } b\}}\right].$$

The competitive guarantee follows simply by combining the above inequalities. □

THEOREM A.2. *For the problem of finding a maximum-weight independent set in a transversal matroid $\mathcal{M}$, Algorithm 6 is 8-competitive in expectation, i.e.,*

$$8 \cdot \mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\mathsf{OPT}\right],$$

where OPT is the weight of a maximum-weight independent set in the transversal matroid $\mathcal{M}$.

Proof. The proof follows through minor, though careful, modifications to the arguments used for Theorem 3.1. We will briefly sketch the main ideas needed.

One can construct the equivalent offline algorithm in a similar manner as described in Algorithm 4. In particular, we first order the set of items I in the same manner as in Algorithm 6. Then, we construct a set A of $2|B|$ *buyer* weights (as opposed to $2|E|$ *edge* weights in Algorithm 4) by adding two values, $a_{b,1}, a_{b,2}$ drawn independently from D_b for each buyer b to A. Then, we iterate over the set A in decreasing order, and for each value a corresponding to a buyer b, we iterate over each edge of the buyer in the predetermined ordering on I, adding at most one edge, using the same decision rule as in Algorithm 4. The equivalence between this offline algorithm and Algorithm 6 follows by the same arguments presented in Claim 3.1.

Employing the same definition of safe edges as in Definition 3.1, one can prove, in the exact same manner as in Lemma 3.1, that for any edge e adjacent to buyer b, and denoting e_b as the edge in E^+ with largest weight,

$$\Pr\left[e_b \text{ is safe for } b \mid e_b = e\right] \geq \frac{1}{2},$$

since the proof of that result relied only on the independence of edge weights for *different* buyers. Just as in Lemma 3.2, the above result immediately implies that

$$\mathbb{E}\left[\sum_{b \in B} r_{e_b} \mathbb{1}_{\{e_b \text{ is safe for buyer } b\}}\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[\sum_{b \in B} r_{e_b}\right].$$

Next, to relate the weight of the set collected by the algorithm with the reward of the safe edges, we note that, as in Lemma 3.3,

$$\mathbb{E}\left[w(M)\right] \geq \mathbb{E}\left[\sum_{b \in B} r_{e_b} \mathbb{1}_{\{e_b \text{ is safe for buyer } b\}}\right],$$

since the proof of this result holds for *any* run of the algorithm (not just in expectation), and relies only on the fact that, if an edge $e_b = \{b, i\}$ is safe for buyer b, then, by construction, either e_b is added to M, or another edge $e' = \{b', i\}$ *that is not safe for buyer b'* with reward $r_{e'} \geq r_{e_b}$ is collected instead.

Finally, since this greedy algorithm is still a 2-approximation to OPT, just as in the proof of Theorem 3.1, it suffices to show that

$$\mathbb{E}\left[\sum_{b \in B} r_{e_b}\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[w(M_S)\right] \geq \frac{1}{4} \cdot \mathbb{E}\left[\mathsf{OPT}\right].$$

The argument here follows closely that presented in the proof of Theorem 2.1. Indeed, consider the first time, in a run of the algorithm, when an edge $e = \{b, i\}$ is being processed and i has not yet been matched in M_S. Then, if the coin flip for buyer b lands heads (with probability $\frac{1}{2}$), then the algorithm will add e to M, and it must be that the item for buyer b in M_S (if any such exists) has smaller weight than r_{e_b}. Otherwise, if the coin flip lands tails, then e_b is added to M_S (and no edge for buyer b will be added to E^+). Thus, denoting e_b as the edge in E^+ with largest weight, it follows that $\mathbb{E}\left[r_{e_b}\right] \geq 1/2 \cdot \mathbb{E}\left[\mathrm{val}_{M_S}(b)\right]$ (where $\mathrm{val}_{M_S}(b)$ denotes the weight of the edge adjacent to buyer b in M_S). Applying linearity of expectation, the claimed bound, and thus the theorem, follows. □

CLAIM 4.1. *The sets E', G_S, and M are distributed the same way when computed by Algorithm 7 as when computed by Algorithm 8.*

Proof. The proof is in the same spirit as Claims 2.1 and 3.1. Indeed, by symmetry, the R-values and S-values follow exactly the same distribution in both Algorithm 7 and Algorithm 8. Further, the construction of the greedy solution G_S in Algorithm 8 depends *only* on the S-values, and thus follows the same distribution as in Algorithm 7. Since M is constructed in the exact same way by both algorithms using the edges E', it suffices to argue, for each fixed realization of edge weights, that E' follows the same distribution in both algorithms.

Now, observe that the checks performed by Algorithm 8 to add an R-value edge $e = \{b, i\}$ to E' exactly correspond to those in Algorithm 7. Indeed, neither algorithm collects an edge with weight smaller than its sample (guaranteed by the threshold τ_i in Algorithm 7, and by the check that $i \in I^S$ in Algorithm 8). Since Algorithm 8 processes the *edges* in decreasing order of weight, item i is removed from I^S at the time when an edge with weight a is assigned to i, which implies that the threshold τ_i for item i will be set to a. Again because of the greedy processing order, the sample capacity $C_S(b)$ when an element $e = \{b, i\}$ with weight a is considered in Algorithm 7 exactly corresponds to $C_S(b, a)$ in Algorithm 8. Hence, it follows that E' follows the same distribution in both algorithms. $\square$