Simple and Optimal Online Contention Resolution Schemes for k-Uniform Matroids

Atanas Dinev ✉
Massachusetts Institute of Technology, Cambridge, MA, USA

S. Matthew Weinberg ✉
Princeton University, NJ, USA

─── **Abstract** ───

We provide a simple $(1 - O(\frac{1}{\sqrt{k}}))$-selectable Online Contention Resolution Scheme for k-uniform matroids against a fixed-order adversary. If A_i and G_i denote the set of selected elements and the set of realized active elements among the first i (respectively), our algorithm selects with probability $1 - \frac{1}{\sqrt{k}}$ any active element i such that $|A_{i-1}| + 1 \le (1 - \frac{1}{\sqrt{k}}) \cdot \mathbb{E}[|G_i|] + \sqrt{k}$. This implies a $(1 - O(\frac{1}{\sqrt{k}}))$ prophet inequality against fixed-order adversaries for k-uniform matroids that is considerably simpler than previous algorithms [2, 4, 18].

We also prove that no OCRS can be $(1 - \Omega(\sqrt{\frac{\log k}{k}}))$-selectable for k-uniform matroids against an almighty adversary. This guarantee is matched by the (known) simple greedy algorithm that selects every active element with probability $1 - \Theta(\sqrt{\frac{\log k}{k}})$ [17].

1 Introduction

Background: OCRSs

Online contention resolution schemes (OCRS), introduced by Feldman et al., are a broadly applicable rounding technique for online selection problems [14]. These are problems in which an algorithm makes irrevocable decisions for whether to select elements arriving online, often subject to combinatorial constraints. Offline, the algorithm knows a distribution over which elements will be "active." Online, elements are revealed to be active or inactive one at a time, and the algorithm must immediately and irrevocably decide whether to accept an active element (inactive elements must be rejected). There are feasibility constraints $\mathcal{F}$ on which elements can be simultaneously accepted. An OCRS for a class of instances is c-selectable if it guarantees that every element is selected with probability at least c, conditioned on being active. See Definition 1 for a formal definition.

Online contention resolution schemes have direct applications to prophet inequalities [14]. In a prophet inequality, a gambler knows the distribution of a sequence of independent random variables $X_1, \ldots, X_n$. Online, each random variable will be sampled one at a time and revealed to the gambler, who immediately and irrevocably decides whether to accept the element, subject to feasibility constraints $\mathcal{F}$. The gambler's goal is to maximize the expected sum of weights of accepted elements, and a prophet inequality compares the ratio of the gambler's expected performance to that of a prophet (who knows all random variables before making decisions). Seminal work of Krengel, Sucheston, and Garling establishes a

tight 1/2-approximation for the single-choice prophet inequality, and seminal work of Samuel-Cahn shows that the same result can be achieved with an especially simple thresholding algorithm [22, 27].

In their work introducing OCRSs, Feldman, Svensson, and Zenklusen prove that a c-selectable OCRS implies a c-approximation for the corresponding prophet inequality [14]. In fact, a c-selectable OCRS provides a c-approximation even to the ex ante relaxation, and Lee and Singla show that OCRSs are *equivalent* to ex ante prophet inequalities [23].

k-Uniform Matroids

k-uniform matroids are a canonical set of feasibility constraints: any set of up to k elements can be selected. Here, a $(1 - O(\sqrt{\frac{\log k}{k}}))$-approximate prophet inequality (whose analysis implicitly extends to a $(1 - O(\sqrt{\frac{\log k}{k}}))$-selectable OCRS) is first developed in [17]. [2] later develops a $(1 - O(\frac{1}{\sqrt{k}}))$-approximation (which also implies a $(1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS), which is tight . [4] further shows how to achieve the same $(1 - O(\frac{1}{\sqrt{k}}))$-approximation using a single sample from each distribution (but their work does not imply any OCRS). [18] tightens the analysis of [2] to nail down exactly the optimal achievable prophet inequality for all k (and this same analysis applies to the implied OCRS). We overview in Section 1.1 these results in more detail, and in particular clarify against what kind of adversary (who selects the order in which the elements are revealed) the guarantees hold.

While the optimal competitive ratio has been known for a decade, and recently tightened to even nail down the precise constants, these algorithms are significantly more complex than Samuel-Cahn's elegant algorithm for the single-choice prophet inequality. For example, [2, 18] both require to solve and analyze a mathematical program in order to accept elements with precisely the correct probability. The rehearsal algorithm of [4] is perhaps simpler, but still requires several lines of pseudocode, and some care with minor details.

Main Result: A Simple, Optimal OCRS

Our main result is a significantly simpler OCRS/prophet inequality for k-uniform matroids that still achieves the optimal guarantee of $1 - O(\frac{1}{\sqrt{k}})$. Of course, it is still not nearly as simple as Samuel-Cahn's single-choice prophet inequality, but a full description fits in two sentences, and the complete analysis is just a few pages.[1] Our OCRS simply denotes by A_i the set of elements it has selected amongst the first i and by G_i the set of realized active elements amongst the first i. Then, when processing element i, we select i with probability $1 - \frac{1}{\sqrt{k}}$ if and only if i is active and $|A_{i-1}| + 1 \leq (1 - \frac{1}{\sqrt{k}}) \cdot \mathbb{E}[|G_i|] + \sqrt{k}$ (otherwise, we discard). Intuitively, our OCRS selects an element if, so far, the number of selected elements does not exceed the expected number of active elements by too much. To turn our OCRS into a prophet inequality, simply let T denote the unique value such that $\sum_i \Pr[X_i > T] = k$.[2] Then, declare X_i to be active if and only if $X_i > T$ and plug this into our OCRS. Compared to prior optimal algorithms for the same setting, our algorithm has the advantage that it is very simple to implement since it does not require solving a complicated dynamic/linear program. (See Section 1.1) We state our algorithm precisely, and prove that it is $(1 - O(\frac{1}{\sqrt{k}}))$-selectable against a fixed-order adversary in Section 4.[3]

---

[1]   The proof does require connecting our algorithm to a random walk, and then analyzing properties of that random walk. But, the proof requires minimal calculations.

[2]   If the distributions have point-masses, smooth them out by adding a uniformly random draw from $[0, \varepsilon]$ for arbitrarily small ε.

[3]   A fixed-order adversary sets the order to reveal the elements offline, and based only on the distributions.

An Impossibility for OCRS against almighty adversary

While the fixed-order adversary is standard in the prophet inequality literature, it is also important to explore the extent to which these same guarantees can hold against an almighty adversary.[4] For example, the prophet inequality of [4] holds against an almighty adversary, but does not imply an OCRS. Our second result shows that this is for good reason: no OCRS can guarantee a selectability better than $1 - \Omega(\sqrt{\frac{\log k}{k}})$ against an almighty adversary. We state and prove this in Section 5.

1.1 Detailed Discussion of Related Work

As previously referenced, prior to our work it is already known that the optimal selectability for OCRS and the optimal prophet inequality against a fixed-order adversary is $1 - \Theta(\frac{1}{\sqrt{k}})$ ([17] proves the impossibility, and [2] designs the first algorithm matching it). [4] designs a prophet inequality that achieves the same $(1 - O(\frac{1}{\sqrt{k}}))$-approximation against an almighty adversary, but this does not imply an OCRS. The analysis in [17] implies an extremely simple OCRS (accept every active element with probability $1 - \Theta(\sqrt{\frac{\log k}{k}})$) that is $(1 - \Theta(\sqrt{\frac{\log k}{k}}))$-selectable against an almighty adversary. We show that this is the best possible guarantee (Theorem 27). Because we view our main result as a simpler algorithm achieving (asymptotically) the same guarantees as prior work, we now overview these works in greater detail.

The γ-Conservative magician in [2]

As previously mentioned, [2] implies an optimal $(1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS against a fixed-order adversary. [2, Definition 2] describes a γ-Conservative Magician, which is an algorithm that adaptively computes thresholds θ_i and accepts an active element on step i if and only if the number of selected elements (or broken wands in the terminology of [2]) W_i in steps $1, \ldots, i-1$ is less than θ_i. The cumulative distribution function of W_i is computed adaptively at every step through a dynamic programming equation. Once the CDF of W_i has been computed, θ_i is chosen so that the ex-ante probability that $W_i \leq \theta_i$ is at least γ. [2] shows that one can choose $\gamma = 1 - \frac{1}{\sqrt{k+3}}$ and $\theta_i \leq k$ for all i, which effectively guarantees an OCRS against a fixed-order adversary that is $1 - \frac{1}{\sqrt{k+3}}$-selectable [2, Theorem 4]. In comparison to [2], our main result achieves the same asymptotic guarantee, but is considerably simpler (in particular, the analysis requires minimal calculations, and there is no dynamic program).

Characterization of the optimal OCRS and prophet inequality for k-uniform matroids in [18]

[18] studies the optimal OCRS for k-uniform matroids (i.e. with optimal selection probability c). They characterize the optimal OCRS for k-uniform matroids as the solution to a linear program. Then, using a differential equation, they show that this optimal solution corresponds to a γ_k^*-Conservative magician, where $\gamma_k^* > 1 - \frac{1}{\sqrt{k+3}}$. The OCRS guarantees in [18] hold against an online adversary [5].

---

[4] An almighty adversary sets the order to reveal online, and with full knowledge of all random variables and all past decisions of the algorithm.

[5] An online adversary adaptively decides which elements to reveal next, based on which elements were active. But the online adversary does not know the status of the unrevealed elements (i.e. it has the same information as the algorithm)

In comparison to [18], our main result achieves the same $1 - O(\frac{1}{\sqrt{k}})$ asymptotic guarantee against a fixed-order adversary, but is, again, considerably simpler and does not require solving any mathematical program. Although our analysis holds against the weaker fixed-order adversary, this assumption is sufficient for many popular applications of OCRS and prophet inequalities in online stochastic optimization.

The Rehearsal algorithm in [4]

As previously discussed, [4] gives an optimal $(1 - O(\frac{1}{\sqrt{k}}))$-approximation prophet inequality against an almighty adversary, and even when knowing only a single sample from each distribution, using the *rehearsal algorithm*. Their rehearsal algorithm takes a sample from each distribution, and stores the $k - 2\sqrt{k}$ highest samples at $T_1, \ldots, T_{k-2\sqrt{k}}$, then repeats $T_i := T_{k-2\sqrt{k}}$ for all $i \in [k - 2\sqrt{k}, k]$. When processing the online element X_e, e is accepted if and only if there is an unfilled slot i with $X_e > T_i$. If X_e is accepted, it fills the highest such slot (the slot with the highest threshold).

Their analysis does not imply an OCRS (indeed, it is not even clear what it would mean to set the thresholds T_i in an OCRS). But, their analysis does hold against an almighty adversary. In comparison, our prophet inequality is simpler, and implies an OCRS. But, our algorithm requires some knowledge of the distributions, rather than just a single sample.[6]

Our analyses have similar flavors: both works connect our algorithms' performance to a random walk. These random walks are quite different (for example, the random walk in [4] is correlated, and ours is not. Our random walk has non-integral step sizes, while theirs does not), and are used to analyze different algorithms. While there are some coincidental similarities (for example, our Lemma A.2 in the full version [10] is a generalization of their Lemma 10), the core of our proof is simply connecting our algorithm to a random walk, whereas the bulk of their proof is coping with the correlation in their random walk and any associated calculations.

Other Related Work

There is substantial additional work on both prophet inequalities and online contention resolution schemes, subject to various other constraints [7, 21, 15, 12, 25, 26, 11, 1, 16, 3, 13, 5, 9, 28, 24]. Aside from this, there is not much technical overlap with these works (in particular, a substantial fraction of these works consider richer feasibility constraints, and therefore achieve constant-factor approximations rather than approximations approaching 1).

Other works have also considered a special class of static threshold policies for k-unit prophet inequalities which set a single threshold and accept any element that exceeds it subject to the feasibility constraint. [17] provides a prophet inequality with a static threshold which is a $1 - O(\sqrt{\frac{\log(k)}{k}})$ approximation. [6] proposes a different static threshold prophet inequality with the same $1 - O(\sqrt{\frac{\log(k)}{k}})$ asymptotic guarantee improving the approximation for small k. [19] uses a mathematical programming approach to show that the policy in [6] is worst-case optimal within all static threshold policies. In contrast to these works, we study the design and limitations of the richer class of adaptive strategies for the more general setting of OCRS.

---

[6]   Turning our OCRS into a prophet inequality requires a value T such that $\sum_i \Pr[X_i > T] \approx k$, and an accurate estimate of $\sum_{j \leq i} \Pr[X_j > T]$ for all i. Estimates up to an additive $\sqrt{k}$ with high probability suffice. This can certainly be achieved with polynomially-many samples, but not a single sample.

In the context of *offline* contention resolution schemes (introduced in [8]), [20] shows a simple optimal contention resolution scheme for k-uniform matroids, which is $(1 - \binom{n}{k}(1 - \frac{k}{n})^{n+1-k}(\frac{k}{n})^k)$-selectable. Their method is not extendable to the online case because it requires knowing the set of active elements A in advance.

1.2 Roadmap

Section 2 follows with preliminaries and definitions. Section 3 is a warm-up that rules out optimal selectability via *extremely* simple greedy algorithms. Section 4 presents a complete proof of our new $(1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS. Section 5 contains a proof of the $1 - \Omega(\sqrt{\frac{\log(k)}{k}})$ upper bound on the probability of selection of any OCRS against almighty adversaries. Section 6 concludes.

2 Preliminaries

2.1 Online contention resolution schemes

Online contention resolution schemes were first introduced by [14] as a broadly applicable online rounding framework. Suppose we are given a finite ground set of elements $N = \{e_1, \ldots, e_n\}$. Consider a family of feasible sets $\mathcal{F} \subset 2^N$ which is downwards-closed (that is, if $I \in \mathcal{F}$ and $J \subseteq I$, then $J \in \mathcal{F}$). Let

$$P_{\mathcal{F}} = \text{conv}(\mathbf{1}_I | I \in \mathcal{F}) \in [0,1]^n$$

be the convex hull of all characteristic vectors in $\mathcal{F}$. We will refer to $P_{\mathcal{F}}$ as the polytope corresponding to the family $\mathcal{F}$.

▶ **Definition 1** (Online contention resolution scheme (OCRS)). *Consider the following online selection setting. A point $x \in P_{\mathcal{F}}$ is given and let $R(x)$ be a random subset of active elements, where element e_i is active with probability x_i. The elements $e \in N$ reveal one by one whether they are active, i.e. $e \in R(x)$, and the decision of the algorithm whether to select an active element is taken irrevocably before the next element is revealed. An OCRS for $P_{\mathcal{F}}$ is an online algorithm that selects a subset $I \subseteq R(x)$ such that $\mathbf{1}_I \in P_{\mathcal{F}}$.*

Many of the natural OCRS considered in [14] are also greedy.

▶ **Definition 2** (Greedy OCRS). *A greedy OCRS π for $P_{\mathcal{F}}$ is an OCRS that for any $x \in P_{\mathcal{F}}$, defines a downwards-closed subfamily of feasible sets $\mathcal{F}_x \subseteq \mathcal{F}$ and an element e is selected when it arrives if, together with the already selected elements, the obtained set is in $\mathcal{F}_x$*

Our next goal is to define the notion of c-selectability, which is a notion of performance of the OCRS. Intuitively, an OCRS is c-selectable if for any element e the probability that the OCRS selects e given that it is active is at least c, where we desire c to be as large as possible. In order to talk about c-selectability in a rigorous way, we need to specify the power of the adversary that chooses the order of the elements revealed to the OCRS in an online fashion (Definition 1). There are three main types of adversaries considered in prior work, which we define below.

▶ **Definition 3** (Strength of adversary). *In the setting of Definition 1, there is an underlying adversary which can choose the order in which the elements are revealed to the OCRS. We define three different types of adversaries:*

- **Offline/Fixed-Order** *adversary, which chooses the order of the elements upfront before any are revealed. Such an adversary knows x and the distribution of $R(x)$, but not the realized active elements.*
- **Online** *adversary, which adaptively chooses next element to reveal using the same information available to the algorithm (x, the distribution of $R(x)$, and which elements have been revealed, which were active, and which were selected).*
- **Almighty** *adversary, which knows upfront the outcomes of all random events, which includes the realization of $R(x)$ and the outcome of the random bits that the OCRS might query.*

We are now ready to define the notion of c-selectability.

▶ **Definition 4** (c-selectability). *Let $c \in [0,1]$. An OCRS for P is c-selectable against an adversary $\mathcal{A}$ if for any $x \in P$ and $e \in N$, we have*

$$\Pr[e \text{ is selected by the OCRS against } \mathcal{A}|e \text{ is active}] \geq c$$

It is often true that a larger probability of selection c can be achieved when x is supposed to be in a down-scaled version of P.

▶ **Definition 5** ((b,c)-selectability). *Let $b, c \in [0,1]$. An OCRS for P is (b,c)-selectable against an adversary $\mathcal{A}$ if for any $x \in b \cdot P$ and $e \in N$, we have*

$$\Pr[e \text{ is selected by the OCRS against } \mathcal{A}|e \text{ is active}] \geq c$$

An important observation is that a (b,c)-selectable OCRS for P against $\mathcal{A}$ implies a bc-selectable OCRS for P against $\mathcal{A}$.

▶ **Observation 6** ([14]). *A (b,c)-selectable OCRS for P implies a bc-selectable OCRS for P.*

The reduction in Observation 6 is as follows: The bc-selectable OCRS essentially runs the given (b,c)-selectable OCRS while scaling down by b each of the probabilities x_i online (i.e. consider selecting an active element independently with probability b). For more details see [14].

▶ Remark 7. It is important to emphasize that a bc-selectable ORCS for P gives selection guarantees on all $x \in P$, while a (b,c)-selectable OCRS for P gives guarantees only when $x \in b \cdot P$ (i.e. a scaled-down version of P).

2.2   OCRS and Prophet inequalities

As previously discussed, one of the many applications of OCRS is to the prophet inequality problem. Here we define the general setting of the prophet inequality problem. We begin with a setup for the environment.

General setting and prophet.   We are given a group set $N = \{e_1, \dots, e_n\}$ and a downwards-closed family $\mathcal{F} \subseteq 2^N$ of feasible subsets. Each of the elements e_i is associated with a value v_i. A prophet is an offline algorithm, which sees the vector $(v_1, \dots, v_n)$ and outputs the feasible set $\mathrm{MAX}(v) = \mathrm{argmax}_{I \in F} \sum_{i \in I} v_i$. We denote by $\mathrm{OPT}(v) = \sum_{i \in \mathrm{MAX}(v)} v_i$ the weight of the maximum set.

▶ **Definition 8** (Prophet inequality). *Suppose we are given a downwards-closed feasibility constraint $\mathcal{F} \subseteq 2^N$. Suppose each element $e_i \in N$ takes value $v_i \in \mathbb{R}_{\geq 0}$ independently from some known distribution $\mathcal{D}_i$. These values are presented one-by-one to an online algorithm π*

in an adversarial order (again specified as offline, online, or almighty). On seeing a value the algorithm needs to immediately and irrevocably decide whether to select the next element e_i, while always maintaining that the set of selected elements so far is in $\mathcal{F}$. Let's denote the set of selected elements by π as $A^*(v)$. We say that π induces a prophet inequality with competitive ratio c for $\mathcal{F}$ if

$$\mathbb{E}_{v \sim \mathcal{D}}[\sum_{i \in A^*(v)} v_i] \geq c \cdot \mathbb{E}_{v \sim \mathcal{D}}[OPT(v)]$$

where $\mathcal{D} = \mathcal{D}_1 \times \ldots \times \mathcal{D}_n$ is the product of the independent distributions $\mathcal{D}_i$

As with OCRS, in order to talk about α-approximation, we need to specify the power of the adversary which specifies the order of the elements revealed. Completely analogously to Definition 3, we could have *offline*, *online*, and *almighty* adversaries.

[14] showed that a c-selectable OCRS against a particular adversary $\mathcal{A}$ implies a c-approximation prophet inequality against an adversary of the same strength.

▶ **Theorem 9** ([14]). *A c-selectable OCRS against (offline/online/almighty) adversary implies the existence of a c-approximation prophet inequality algorithm.*

In the classical prophet inequality formulation [22], the value of the online algorithm π is compared directly to the offline optimum. [23] consider an ex-ante prophet inequality, where the value of π is compared to the optimal value of a convex relaxation, which upper bounds the offline optimum. [23] show that this stronger notion of an ex-ante prophet inequality is equivalent to an OCRS.

2.3 k-uniform matroids

In this section, we give a definition for k-uniform matroids, which is the feasibility constraint that we will use throughout the paper. Given a ground set $N = \{e_1, \ldots, e_n\}$, the k-uniform matroid is the matroid consisting of all subsets of N of size at most k.

▶ **Definition 10** (k-uniform matroid). *The k-uniform matroid for N is $M_k = (N, \mathcal{F}_k)$, where*

$$\mathcal{F}_k = \{S \subseteq N | |S| \leq k\}$$

and the corresponding polytope of $\mathcal{F}_k$ is given by

$$P_k = \{x \in \mathbb{R}^n_{\geq 0} | \sum_{i=1}^n x_i \leq k\}$$

We remind the reader of prior work on OCRSs and prophet inequalities for k-uniform matroids below.

▶ **Theorem 11** ([17, 2, 4, 18]). *The following is known, prior to our work, on OCRSs and prophet inequalities for k-uniform matroids:*
- *Against a fixed-order/online adversary, the best prophet inequalities and OCRSs for $\mathcal{F}_k$ achieve a guarantee of $1 - \Theta(\frac{1}{\sqrt{k}})$ (lower bound: [17], algorithm: [2, 18]).*
- *Against an almighty adversary, the best prophet inequalities for $\mathcal{F}_k$ achieve a guarantee of $1 - \Theta(\frac{1}{\sqrt{k}})$ (lower bound: [17], algorithm: [4]).*
- *Against an almighty adversary, the best-known OCRS achieves a guarantee of $1 - \Theta(\sqrt{\frac{\log k}{k}})$ (implicit in [17]).*

3   Warmup: Naive approaches towards an OCRS

The goal of this section is to explore a few exceptionally simple algorithms that one might try to use to construct an optimal OCRS for k-uniform matroids. We will present results about whether optimal factors are possible against adversaries of variable strengths. In particular, we will show that one cannot achieve a $(1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS by using a very simple greedy algorithm or a tweaked variant of it utilizing a partition matroid.

We first consider a naive greedy OCRS, which greedily selects active elements until it has selected k elements. Formally in the language of Definition 2, for this OCRS we have $\mathcal{F} = \{S \subseteq N||S| = k\}$ and $\mathcal{F}_x = \mathcal{F}$ for all x. We quickly establish that the naive greedy OCRS is not (b, c)-selectable even against an offline/fixed-order adversary for any b, c satisfying $bc = 1 - O(\frac{1}{\sqrt{k}})$. This would rule out constructing exceptionally simple OCRS via Observation 6.

▶ **Theorem 12.** *There are no b, c, satisfying $bc = 1 - O(\frac{1}{\sqrt{k}})$, such that the naive greedy OCRS is (b, c)-selectable against the offline adversary.*

Proof. See Appendix A.1 in full version [10] for a proof.   ◀

Our second result is that even if we complicate the naive greedy OCRS slightly it does not imply an optimal factor. Suppose instead of using the k-uniform matroid, we use a partition matroid and the algorithm is to greedily select active elements as long as the set of selected elements lies in the partition matroid. Formally in the language of Definition 2, the feasibility family $\mathcal{F}_x$ is given by a partition matroid (which could depend on x). For a given x, the partition matroid is of the form $\{(n_i, k_i, S_i)\}_{i=1}^s$, where $\sum_{i=1}^s k_i = k$, $\sum_{i=1}^s n_i = n$ and the sets S_i are pairwise disjoint and satisfy $\cup_{i=1}^s S_i = N$ and $|S_i| = n_i$. Here we can select at most k_i elements from S_i. We next prove that such scheme is not (b, c)-selectable even against the *offline* adversary for any b, c satisfying $bc = 1 - O(\frac{1}{\sqrt{k}})$.

▶ **Theorem 13.** *There are no b, c, satisfying $bc = 1 - O(\frac{1}{\sqrt{k}})$, such that the naive greedy OCRS with a partition matroid is (b, c)-selectable against the offline adversary.*

Proof. See Appendix A.1 in full version [10] for a proof.   ◀

▶ Remark 14. Theorems 12 and 13 say that one cannot use the transformation in Observation 6 on a naive greedy OCRS to obtain a $(1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS. Thus ruling out some exceptionally simple ways to construct a $(1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS for k-uniform matroids.

▶ Remark 15. Intuitively, the above variations of the naive greedy schemes fail to be optimal because they tend to select too many elements early in the process in comparison to the expected number of active elements so far. Thus, they are likely to run out of space when they reach the last element. Our main algorithm in Section 4 attempts to counter this.

We conclude by reminding the reader that the naive greedy OCRS is (b, c)-selectable against the almighty adversary for some b, c satisfying $bc = 1 - O(\sqrt{\frac{\log(k)}{k}})$. The proof is implicit in the analysis of the $(1 - O(\sqrt{\frac{\log(k)}{k}}))$-approximate prophet inequality of [17].

▶ **Theorem 16** (Implicit in [17]). *The naive greedy OCRS is $(1 - \sqrt{\frac{2\log(k)}{k}}, 1 - \frac{1}{k})$-selectable against the almighty adversary. By the transformation in Observation 6 this implies a $(1 - O(\sqrt{\frac{\log(k)}{k}}))$-selectable OCRS for k-uniform matroids.*

Proof. We remind the reader of the simple proof in Appendix A.1 in our full version [10].   ◀

4 A simple optimal OCRS for k-uniform matroids

The goal of this section is to give a new $(1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS against offline/fixed-order adversaries. Because the adversary must commit to an ordering using just knowledge of x, and the distribution of $R(x)$, we let $e_1, e_2 \ldots, e_n$ refer to the elements that are revealed, in order. Note that the events $e_i \in R(x)$ are independent. We will show that the following algorithm is a $(1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS for k-uniform matroids.

OCRS(x):
1. Initialize the set of selected elements $A_0 = \emptyset$.
2. For $i = 1, \ldots, n$ do:
   2.a If e_i is active and $|A_{i-1}| + 1 \leq (1 - \frac{1}{\sqrt{k}})(\sum_{j \leq i} x_j) + \sqrt{k}$, then select e_i with probability $(1 - \frac{1}{\sqrt{k}})$ (i.e. $A_i = A_{i-1} \cup e_i$) and otherwise discard it (i.e. $A_i = A_{i-1}$).
   2.b If $|A_{i-1}| + 1 > (1 - \frac{1}{\sqrt{k}})(\sum_{j \leq i} x_j) + \sqrt{k}$ or e_i is inactive, then discard e_i (i.e. $A_i = A_{i-1}$).

Observe that **OCRS** is derived from an even simpler $(1 - \frac{1}{\sqrt{k}}, 1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS, by using the reduction of Observation 6. We clearly state this simpler algorithm below (we also state it parameterized by d, as our entire analysis follows for general d, and then is optimized for $d := \sqrt{k}$ at the very end).

Algorithm(d, x):
1. Initialize the set of selected elements $B_0 = \emptyset$.
2. For $i = 1, \ldots, n$ do:
   2.a If e_i is active and $|B_{i-1}| + 1 \leq \sum_{j \leq i} x_j + d$, then select e_i, and otherwise discard it.
   2.b If $|B_{i-1}| + 1 > \sum_{j \leq i} x_j + d$ or e_i is inactive, then discard e_i.

▶ **Observation 17.** *If **Algorithm**($\sqrt{k}, x$) is a $(1 - \frac{1}{\sqrt{k}}, 1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS for P_k, then **OCRS**(x) is a $(1 - O(\frac{1}{\sqrt{k}}))$-selectable OCRS for P_k.*

Proof. **OCRS**(x) is exactly the result of applying the [14] reduction of Observation 6 to **Algorithm**($\sqrt{k}, x$), with $b = 1 - \frac{1}{\sqrt{k}}$ and $c = 1 - O(\frac{1}{\sqrt{k}})$. Therefore, by Observation 6, **OCRS**(x) is $(1 - \frac{1}{\sqrt{k}}) \cdot (1 - O(\frac{1}{\sqrt{k}}))$-selectable (i.e. $(1 - O(\frac{1}{\sqrt{k}}))$-selectable) whenever **Algorithm**($\sqrt{k}, x$) is $(1 - \frac{1}{\sqrt{k}}, 1 - O(\frac{1}{\sqrt{k}}))$-selectable. ◄

In line with Remark 7, we emphasize that **Algorithm**($\sqrt{k}, x$) operates in a universe where the probabilities x_i are scaled down by $b = 1 - \frac{1}{\sqrt{k}}$, while **OCRS**($x$) operates with the original probabilities x_i.

Our key proposition is that **Algorithm**(d, x) is indeed sufficiently selectable.

▶ **Proposition 18.** ***Algorithm**(d,x) is $(1 - \frac{d}{k}, 1 - \frac{2}{d-1})$-selectable over P_k. That is, for all $x \in (1 - \frac{d}{k}) \cdot P_k$, **Algorithm**(d,x) is $(1 - \frac{2}{d-1})$-selectable.*

The proof of Proposition 18 proceeds in two steps. The first (shorter) step is to guarantee that **Algorithm**(d, x) always selects at most k elements. The second is to show that every element is accepted with sufficient probability.

▶ **Observation 19.** *For all $x \in (1 - \frac{d}{k}) \cdot P_k$, **Algorithm**(d, x) accepts at most k elements.*

Proof. Observe that, at all times, $|B_i| \leq \sum_{j \leq i} x_j + d$. As $\sum_j x_j \leq k \cdot (1 - \frac{d}{k}) = k - d$, this implies that $|B_n| \leq k$, and the algorithm accepts at most k total elements. ◄

We now proceed to prove that the algorithm is sufficiently selectable. For this part of the analysis, it will be convenient to consider the process $S_i := |B_i| - \sum_{j \leq i} x_j$. Observe that S_i has the following dynamics. First, $S_0 = 0$. Further if $S_{i-1} + 1 - x_i \leq d$ (i.e. have "space" to accept e_i), then

$$S_i := \begin{cases} S_{i-1} + 1 - x_i & \text{if } e_i \text{ is active (with probability } x_i) \\ S_{i-1} - x_i & \text{if } e_i \text{ is inactive (with probability } 1 - x_i) \end{cases}$$

and if $S_{i-1} + 1 - x_i > d$ (i.e. don't have "space" to accept e_i), then $S_i = S_{i-1} - x_i$ regardless of whether e_i is active (with probability 1). Additionally, consider the process $W_i = \sum_{j=1}^{i} X_j$, where

$$X_i = \begin{cases} 1 - x_i & \text{if } e_i \text{ is active (with probability } x_i) \\ -x_i & \text{if } e_i \text{ is inactive (with probability } 1 - x_i) \end{cases}$$

Intuitively, W_i tracks the number of *active* elements above expectation (among the first i), and S_i tracks the number of *selected* elements above the expected number of active elements (among the first i).

▶ **Lemma 20.** *$W_i - S_i$ is exactly equal to the number of active elements that are discarded amongst the first i.*

Proof. Let G_i be the set of active elements in the first i revealed elements. Then, by definition $W_i = |G_i| - \sum_{j \leq i} x_j$. Thus, $W_i - S_i = |G_i| - |B_i|$ and the conclusion follows because $|B_i|$ is the number of active and selected elements in the first i revealed ones. ◀

By Lemma 20 it follows that the difference $W_i - S_i$ increases by one if and only if e_i is active and discarded, and stays the same otherwise. The rest of the proof requires just two more natural steps. First, we characterize which elements are active and discarded, just as a property of the random process W. Second, we bound the probability of this occurring.

▶ **Lemma 21.** *Element e_i is active and discarded by **Algorithm**(d, x) if and only if $\lceil W_i \rceil > d$ and $\lceil W_i \rceil > \lceil W_j \rceil$ for all $j < i$. That is, e_i is active and discarded by **Algorithm**(d, x) if and only if the random process W reaches a new integral height for the first time.*

Proof. By Lemma 20 the difference $W_i - S_i$ increases by 1 when e_i is active and discarded and stays the same otherwise. Therefore, the n-th active and discarded elements is e_i, where i is the smallest index such that $W_i - S_i = n$. Let's now fix an arbitrary $n \geq 1$. To prove the lemma, it is enough to show that i is the smallest index such that $W_i - S_i = n$ if and only if i is the smallest index such that $\lceil W_i \rceil = d + n$.

We begin with the "only if" direction. Suppose i_n is the smallest index such that $W_{i_n} - S_{i_n} = n$. By Lemma 20, element e_{i_n} is active and discarded. Therefore, by the selection rule of **Algorithm**(d, x) we get that $S_{i_n - 1} + 1 - x_{i_n} > d$ and therefore $S_{i_n} = S_{i_n - 1} - x_{i_n}$. This implies that $S_{i_n} > d - 1$. We also know that $S_{i_n} \leq d$ (because S_i stores the difference between $|B_i|$ and $\sum_{j \leq i} x_i$, which is hard-coded to be at most d). Using these two inequalities along with our hypothesis that $W_{i_n} - S_{i_n} = n$ we obtain:

$$W_{i_n} = S_{i_n} + n \in (d + n - 1, d + n], \text{ and therefore: } \lceil W_{i_n} \rceil = d + n.$$

Further, by definition of i_n as the *first* index such that $W_i - S_i = n$, we know that for all $j < i_n$: $W_j \leq S_j + n - 1$. Moreover, we also know that $S_j \leq d$ for all j (again, because S_j stores the difference between $|B_j|$ and $\sum_{\ell \leq j} x_\ell$, which is at most d). Therefore, $W_j \leq d + n - 1$,

and we conclude that $\lceil W_j \rceil \leq d + n - 1 < \lceil W_{i_n} \rceil$ for all $j < i_n$. Therefore, i_n is the smallest index such that $\lceil W_{i_n} \rceil = d + n$. This establishes that if i_n is the smallest index such that $W_{i_n} - S_{i_n} = n$, then i_n is the smallest index such that $\lceil W_{i_n} \rceil = d + n$.

Now we show the "if" direction. Suppose that i_n is the smallest index such that $\lceil W_{i_n} \rceil = d + n$. Since $W_{i_n} - S_{i_n}$ is an integer, and because $S_{i_n} \leq d$, it must be the case that $W_{i_n} - S_{i_n} \geq n$. Let $i^* \leq i_n$ be the smallest index such that $W_{i^*} - S_{i^*} = n$ (such i^* exists because $W_i - S_i$ always increases by 1 or stays the same, and because we have just shown that $W_{i_n} - S_{i_n} = n$). By the "only if" proof above, i^* is the smallest index such that $\lceil W_{i^*} \rceil = d + n$, implying that in fact $i^* = i_n$, as desired. This establishes that if i_n is the smallest index such that $\lceil W_{i_n} \rceil = d + n$, then i_n is also the smallest index such that $W_{i_n} - S_{i_n} = n$.

This completes the proof: the nth active element discarded by the algorithm is the smallest index such that $W_i - S_i = n$. By the work above, this is exactly the smallest index such that $\lceil W_i \rceil = n + d$. Therefore, discarded elements are exactly those that reach a new integral height for the first time. ◄

Our remaining task is simply to upper bound the probability that W_i reaches a new integral height, for all i.

▶ **Lemma 22.** *For each $m \in \{1, \ldots, n\}$*

$$\Pr(\lceil W_m \rceil > d \text{ and } \lceil W_m \rceil > \lceil W_i \rceil \text{ for } i < m) \leq \frac{2x_m}{d-1}$$

Proof. Fix $m \in \{1, \ldots, n\}$. Consider the process $Q_i := W_{m-i-1} - W_{m-1}$ for $i = 0, 1, \ldots, m-1$. Intuitively, $\{Q_i\}_{i=0}^{m-1}$ is the "reversed" process W starting at time-step $m - 1$. Note that $Q_0 = 0$ and $Q_{m-1} = -W_{m-1}$. Observe also that

$$Q_i - Q_{i-1} = W_{m-i-1} - W_{m-1} - (W_{m-i} - W_{m-1}) = W_{m-i-1} - W_{m-i} = -X_{m-i}$$

Note that since the adversary does not see which elements are active and has to commit to their order a priori, we know that $X_1, \ldots, X_n$ are independent. We also note that

$$\mathbb{E}[X_i] = x_i(1 - x_i) - x_i(1 - x_i) = 0$$

By combining the previous two facts, we obtain that $\{Q_i\}_{i=0}^{m-1}$ is a discrete martingale. Let's denote its maximum by $M_{m-1} = \max_{1 \leq i \leq m-1} Q_i$. We will next show that if $\lceil W_m \rceil > d$ and $\lceil W_m \rceil > \lceil W_i \rceil$ for $i < m$, then the following two events have to hold:
- e_m is active.
- $M_{m-1} < 1$ and $Q_{m-1} \leq -(d-1)$. That is, the martingale Q can never reach a height of 1, *and* it must finish below $-(d-1)$.

Indeed, since $\lceil W_m \rceil > \lceil W_{m-1} \rceil$, then e_m is active. Also since $\lceil W_m \rceil > \lceil W_i \rceil$ for $i < m$, we know that $W_m > W_{m-i-1}$ for all i. Combining this with the inequality $W_m \leq W_{m-1} + 1$, we obtain

$$Q_i = W_{m-i-1} - W_{m-1} \leq W_{m-i-1} - W_m + 1 < 1$$

Thus, $Q_i < 1$ for all $i \in [1, m-1]$ or equivalently $M_{m-1} < 1$. The condition $\lceil W_m \rceil > d$ implies that $W_m > d$. Using the last inequality we obtain

$$Q_{m-1} = -W_{m-1} \leq -W_m + 1 \leq -(d-1)$$

Therefore, we obtained that e_m is active, $M_{m-1} < 1$ and $Q_{m-1} \leq -(d-1)$ as desired.

Using the above property combined with the fact that the event whether e_m is active is independent of the events $M_{m-1} < 1$ and $Q_{m-1} \le -(d-1)$ (because M_{m-1} and Q_{m-1} are determined entirely by the previous $m-1$ elements) we obtain that:

$$\Pr(\lceil W_m \rceil > d \text{ and } \lceil W_m \rceil > \lceil W_i \rceil \text{ for } i < m)$$
$$\le \Pr(\{e_m \text{ is active}\} \cap \{M_{m-1} < 1 \text{ and } Q_{m-1} \le -(d-1)\})$$
$$= \Pr(e_m \text{ is active}) \cdot \Pr(M_{m-1} < 1 \text{ and } Q_{m-1} \le -(d-1))$$
$$= x_m \cdot \Pr(M_{m-1} < 1 \text{ and } Q_{m-1} \le -(d-1))$$

So, the final step is to upper bound $\Pr(M_{m-1} < 1 \text{ and } Q_{m-1} \le -(d-1))$, which is just a claim about martingales that change by at most 1 in each step. Lemma A.2 in [10] (full version), which is a short application of the Optional Stopping Theorem, applied for $a = 1, b = -(d-1) < 0$, and $K = 1$ implies that:

$$\Pr(M_{m-1} < 1 \text{ and } Q_{m-1} \le -(d-1)) \le \frac{2}{d-1}.$$

Therefore,

$$\Pr(\lceil W_m \rceil > \lceil W_i \rceil \text{ for } i < m \text{ and } \lceil W_m \rceil > d) \le \frac{2x_m}{d-1},$$

which concludes the proof of the lemma.   ◀

This suffices to wrap up the proof of Proposition 18.

Proof of Proposition 18. We have that

$$\Pr(e_m \text{ is selected}|e_m \text{ is active}) = 1 - \Pr(e_m \text{ is discarded}|e_m \text{ is active})$$
$$= 1 - \frac{\Pr(e_m \text{ is active and discarded })}{\Pr(e_m \text{ is active})}$$
$$\ge 1 - \frac{\frac{2x_m}{d-1}}{x_m} = 1 - \frac{2}{d-1} \text{ (by Lemma 22).} \qquad ◀$$

Setting $d = \sqrt{k}$ in Proposition 18 we get

▶ **Corollary 23.** *$\boldsymbol{Algorithm}(\sqrt{k}, x)$ is $(1 - \frac{1}{\sqrt{k}}, 1 - \frac{2}{\sqrt{k}-1})$-selectable, and therefore $\boldsymbol{OCRS(x)}$ is $(1 - O(\frac{1}{\sqrt{k}}))$-selectable.*

▶ Remark 24. An intuitive comparison with the extremely simple OCRS implied in [17] is the following. The OCRS from [17] needs to "scale" probabilities down with $(1 - \Theta(\sqrt{\frac{\log(k)}{k}}))$ in order to have a $(1 - \Theta(\sqrt{\frac{\log(k)}{k}}))$ chance of not running out of space. Our result shows that if we adaptively select active elements whenever the number of selected ones does not exceed the expected number by d, then we only need to "scale" down probabilities by $(1 - \Theta(\frac{1}{\sqrt{k}}))$ to have a $(1 - \Theta(\frac{1}{\sqrt{k}}))$ chance of not running out of space.

▶ Remark 25. In terms of exact constants, Corollary 23, gives that $\boldsymbol{OCRS}(x)$ is $(1 - \frac{4}{\sqrt{k}})$-selectable for $k \ge 25$.

▶ Remark 26. Intuitively, an online adversary can potentially manipulate the algorithm by revealing an element e_j with a large (resp. small) x_j when the algorithm has selected many (resp. few) elements above the expectation. In this scenario, the proof of Lemma 22 will not go through since the "reversed" process Q_i could have correlated steps and will (in general) fail to be a martingale.

5 Upper bound against almighty adversaries

Recall that [17] implies that the naive greedy OCRS can be used to obtain a $(1 - O(\sqrt{\frac{\log(k)}{k}}))$-selectable OCRS against the almighty adversary. (Theorem 16) We recall that the almighty adversary knows which elements are active a priori and also knows everything about the OCRS (Definition 3). In this section we show that, against the almighty adversary, the probability of selection of any OCRS cannot be greater than $(1 - \Omega(\sqrt{\frac{\log(k)}{k}}))$. This implies that, against almighty adversaries the factor of $(1 - O(\sqrt{\frac{\log(k)}{k}}))$, achieved by the naive greedy OCRS, is asymptotically optimal. In particular, our main goal in this section is to prove the following theorem.

▶ **Theorem 27.** *Suppose that a c-selectable OCRS for the k-uniform matroid against the almighty adversary exists. Then $c \leq 1 - \Omega(\sqrt{\frac{\log(k)}{k}})$*

In our proof of Theorem 27 we will only consider the instance $n = 2k$ and vector of probabilities $x_i = \frac{1}{2}$ for all $i \in [1, 2k]$ (i.e. each element is active with probability $\frac{1}{2}$). To execute the proof, we will use the following strategy.

- We will consider the following subclass of almighty adversaries. The adversary knows which elements are active and everything about the OCRS. However, it needs to:
  - commit to the order in which the elements will be revealed a priori
  - reveal all active elements before all inactive ones

  This type of restriction will be convenient for the analysis. We will refer to the class of such adversaries as $\mathcal{H}$.
- Assuming the existence of a c-selectable OCRS π against adversaries of class $\mathcal{H}$, we show that there exists c-selectable OCRS π^s against adversaries of class $\mathcal{H}$, which selects the i-th revealed active element with probability independent of the identities of the first i revealed elements. In other words, the probability that π^s selects the i-th revealed active element is a function $g(i)$ (which depends on π but not on the identities of the first i revealed elements). (Section 5.1)
- By using the probability values $\{g(i)\}_{i=1}^{2k}$, we construct an adversary in $\mathcal{H}$. Based on this adversary, we upper bound the probability of selection c by using the solution to a linear programming relaxation. (Section 5.2)

Before we proceed with the proof we will fully describe a general model for how an OCRS works. An arbitrary OCRS π operates in the following way:

1. Before any elements are revealed π can flip some random coins $coins_1$.
2. Once the first element a_1 is revealed (and whether it is active or not), π can flip more random coins $coins_1'$ and it makes a decision to select / discard a_1 with some probability based on $coins_1$, $coins_1'$ and the identity of the element a_1. Let b_1 be the indicator random variable of the event that a_1 is selected.
3. Based on the history so far, i.e. $coins_1, a_1, coins_1', b_1$, it flips more random coins $coins_2$ before the second element is revealed.
4. After the identity of the second element a_2 (and its activity) is revealed π flips more coins $coins_2'$ (based on the history so far), and makes a decision to select a_2 or not, which is recorded in an indicator variable b_2.
5. Let $B_i = (coins_i, a_i, coins_i', b_i)$ where $coins_i$ is the random coins π flips right before seeing the i-the revealed element a_i, $coins_i'$ is the random coins π flips after seeing a_i, and b_i is the indicator of the event that π selects a_i.

6. In general, as a function of the history $\overline{B}_i = \{B_1, \ldots, B_i\}$, π flips random coins $coins_{i+1}$. Then the $(i+1)$-th element a_{i+1} is revealed. Based on the history $\overline{B}_i \cup coins_{i+1} \cup a_{i+1}$, π flips more random coins $coins'_{i+1}$. Finally, based on the history $\overline{B}_i \cup coins_{i+1} \cup a_{i+1} \cup coins'_{i+1}$, π decides to select / discard a_{i+1}, which generates the indicator b_{i+1}.

7. The procedure described in **6.** is repeated until the n-th element is selected / discarded.

5.1   Defining a symmetric OCRS

Suppose we have a c-selectable OCRS π. Our goal in this section will first be to define what it means to "apply a permutation" to π. Then we will define the "symmetric" OCRS π^s in the following way:

1. Sample uniformly random permutation σ of N.

2. Apply σ to π.

We will then show that the probability that π^s selects the i-th active revealed element is independent on the identities of the first i revealed elements.

We will first introduce some definitions. Given an OCRS π, and a permutation σ of the ground set N, we define π_σ as the OCRS which "treats" each element a_i exactly like π would "treat" $\sigma^{-1}(a_i)$. Formally we use the following definition.

▶ **Definition 28.** *Let π be an OCRS and σ a permutation of N. Define π_σ as an OCRS which uses π as a black-box in the following way:*

1. *Before any elements are revealed π_σ queries π to flip some random coins $coins_1$.*

2. *Once the first element a_1 is revealed (and whether it is active or not), π_σ queries π on $(coins_1, \sigma^{-1}(a_1))$ to flip more random coins $coins'_1$. Based on $(coins_1, \sigma^{-1}(a_1), coins'_1)$, π will make some decision to select / discard $\sigma^{-1}(a_1)$ with some probability. Then π_σ selects a_1 if and only if π selects $\sigma^{-1}(a_1)$. Let b_1 be the indicator random variable of the event that a_1 is selected by π_σ.*

3. *π_σ queries π on history $coins_1, \sigma^{-1}(a_1), coins'_1, b_1$ to generate random coins $coins_2$ before the second element is revealed.*

4. *After the identity of the second element a_2 (and its activity) is revealed π_σ queries π on $coins_1$, $\sigma^{-1}, (a_1), coins'_1, b_1, coins_2, \sigma^{-1}(a_2)$ to flip more random coins $coins'_2$. Then it queries π on $coins_1$, $\sigma^{-1}(a_1)$, $coins'_1, b_1, coins_2, \sigma^{-1}(a_2), coins'_2$ whether to select / discard $\sigma^{-1}(a_2)$, and π_σ selects a_2 if and only if π selects $\sigma^{-1}(a_2)$.*

5. *Let $B_i = (coins_i, a_i, coins'_i, b_i)$ where $coins_i$ is the random coins π_σ flips by querying π right before seeing the i-the revealed element a_i, $coins'_i$ is the random coins π_σ flips by querying π after seeing a_i, and b_i is the indicator of the event that π_σ selects a_i.*

6. *In general, as a function of the history $\overline{B}_i = \{B_1, \ldots, B_i\}$, π_σ will query π on $\overline{B}'_i = \{B'_1, \ldots, B'_i\}$, where $B'_i = (coins_i, \sigma^{-1}(a_i), coins'_{i+1}, b_i)$ to flips random coins $coins_{i+1}$ before the $(i+1)$-th active element a_{i+1} is revealed. Then it queries π on $\overline{B}'_i \cup coins_{i+1} \cup \sigma^{-1}(a_{i+1})$, to generate more random coins $coins'_{i+1}$, and then it queries it again on $\overline{B}'_i \cup coins_{i+1} \cup \sigma^{-1}(a_{i+1}) \cup coins'_{i+1}$ whether to select a_{i+1} if and only if π selects $\sigma^{-1}(a_{i+1})$. This generates an indicator b_{i+1}.*

7. *The procedure described in **6.** is repeated until the n-th element is selected / discarded.*

Similarly to applying a permutation to an OCRS π, we can apply a permutation to an adversary $\mathcal{A}$, resulting in an adversary $\mathcal{A}_\sigma$. Intuitively, $\mathcal{A}_\sigma$ "treats" element a like $\mathcal{A}$ would treat $\sigma^{-1}(a)$. We give the following formal definition. Recall that $\mathcal{H}$ is the class of adversaries defined in the beginning of this section.

▶ **Definition 29.** *Let $\mathcal{A} \in \mathcal{H}$ be an adversary and σ a permutation of the ground set N. We define the adversary $\mathcal{A}_\sigma$ as operating against an OCRS π in the following way:*

1. *Given a set of active elements A, $\mathcal{A}_\sigma$ queries $\mathcal{A}$ on a set of active elements $\sigma^{-1}(A)$ against the OCRS $\pi_{\sigma^{-1}}$. Upon this query $\mathcal{A}$ returns an order in which to reveal the elements from N.*

2. *Given this order, if $\mathcal{A}$ chose to reveal element a in the i-th position, $\mathcal{A}_\sigma$ reveals $\sigma(a)$ in the i-th position.*

▶ **Remark 30.** It is easy to see that if $\mathcal{A} \in \mathcal{H}$, then $\mathcal{A}_\sigma \in \mathcal{H}$.

We next show that the operation of applying a permutation to an OCRS or adversary is invertible.

▶ **Proposition 31.** *For a given permutation σ, the maps $\pi \to \pi_\sigma$ and $\mathcal{A} \to \mathcal{A}_\sigma$ are bijections, with inverses $\pi \to \pi_{\sigma^{-1}}$ and $\mathcal{A} \to \mathcal{A}_{\sigma^{-1}}$ respectively.*

Proof. See Appendix A.3 in full version [10] for a proof. ◀

We will now need the following helpful lemma.

▶ **Lemma 32.** *Let $a \in N$ be an element and $A \subseteq N$ a subset of elements. Let σ be a permutation of N, π an OCRS, and $\mathcal{A} \in \mathcal{H}$ an adversary. Then*

$$\Pr(\pi \text{ selects } a \text{ against } \mathcal{A} | A \text{ are active}) = \Pr(\pi_\sigma \text{ selects } \sigma(a) \text{ against } \mathcal{A}_\sigma | \sigma(A) \text{ are active})$$

Proof. Consider the interaction between π_σ and $\mathcal{A}_\sigma$ on set of active elements $\sigma(A)$. By Definition 29 before the process begins $\mathcal{A}_\sigma$ queries $\mathcal{A}$ on $\sigma^{-1}(\sigma(A)) = A$ against OCRS $(\pi_\sigma)_{\sigma^{-1}} = \pi$ (Proposition 31). Based on this if $\mathcal{A}$ chooses to first reveal the active elements A to π in some order, $\mathcal{A}$ will reveal $\sigma(A)$ to π_σ in the same order (Definition 29). Thus, if we now pair π and π_σ as in Definition 28, we know that in the interaction when π is given history $\overline{B}$ and π_σ will be given history $\overline{B'}$, obtained from $\overline{B}$ by replacing each element b by $\sigma(b)$. Thus, by Definition 28 it follows that π_σ will select element $\sigma(a)$ if and only if π selects element a. This finishes the proof. ◀

We are now ready to show if π is c-selectable against adversaries in $\mathcal{H}$, then π_σ is also c-selectable against adversaries in $\mathcal{H}$.

▶ **Lemma 33.** *If π is c-selectable against adversaries in $\mathcal{H}$, then π_σ is also c-selectable against adversaries in $\mathcal{H}$.*

Proof. Let $\mathcal{A} \in \mathcal{H}$ be an adversary and $a \in N$ an arbitrary element. We have that

$\Pr(\pi_\sigma \text{ selects } a \text{ against } \mathcal{A}) =$

$\displaystyle = \sum_{A \subseteq N} \Pr(\pi_\sigma \text{ selects } a \text{ against } \mathcal{A} | A \text{ are active}) \Pr(A \text{ are active})$

$\displaystyle = \sum_{A \subseteq N} \Pr(\pi_\sigma \text{ selects } a \text{ against } \mathcal{A} | A \text{ are active}) \frac{1}{2^{2k}}$

$\displaystyle = \sum_{A \subseteq N} \Pr(\pi \text{ selects } \sigma^{-1}(a) \text{ against } \mathcal{A}_{\sigma^{-1}} | \sigma^{-1}(A) \text{ are active}) \frac{1}{2^{2k}}$ (Lemma 32, Prop. 31)

$\displaystyle = \sum_{A \subseteq N} \Pr(\pi \text{ selects } \sigma^{-1}(a) \text{ against } \mathcal{A}_{\sigma^{-1}} | \sigma^{-1}(A) \text{ are active}) \Pr(\sigma^{-1}(A) \text{ are active})$

$\displaystyle = \Pr(\pi \text{ selects } \sigma^{-1}(a) \text{ against } \mathcal{A}_{\sigma^{-1}}) \geq \frac{c}{2}$ (by c-selectability of π)

where in the last equality we used that $x_i = \frac{1}{2}$ for all i. Thus,

$$\Pr(\pi_\sigma \text{ selects } a \text{ against } \mathcal{A} | \ a \text{ is active}) = \frac{\Pr(\pi_\sigma \text{ selects } a \text{ against } \mathcal{A})}{\Pr(a \text{ is active})} \geq c$$

since $\Pr(a \text{ is active}) = \frac{1}{2}$. ◀

▶ **Remark 34.** Notice that in the proof of Lemma 33 it was crucial that $x_i = \frac{1}{2}$ for all i, which we used to claim $\Pr(A \text{ are active}) = \Pr(\sigma^{-1}(A) \text{ are active}) = \frac{1}{2^{2k}}$.

Suppose we have a c-selectable OCRS π against adversaries in $\mathcal{H}$. We define the "symmetric" OCRS π^s in the following way:

1. Sample a uniformly random permutation σ of the ground set N.
2. Operate like π_σ

We will next show that π^s is c-selectable against adversaries in $\mathcal{H}$ and that it does not differentiate between identities of different elements.

▶ **Lemma 35.** *The OCRS π^s is c-selectable against adversaries in $\mathcal{H}$.*

Proof. Let $\mathcal{A} \in \mathcal{H}$ be an adversary. Since $\mathcal{A}$ needs to decide on the order in which to reveal the elements before seeing what permutation σ is drawn in step **1.**, we know that the order of elements revealed does not depend on σ. Thus, for an arbitrary element a we have that

$$\Pr(\pi^s \text{ selects } a \text{ against } \mathcal{A}) = \sum_\sigma \Pr(\pi_\sigma \text{ selects } a \text{ against } \mathcal{A}) \frac{1}{n!} \geq \text{ (by Lemma 33)}$$

$$\geq \sum_\sigma \frac{c}{2} \frac{1}{n!} = \frac{c}{2}$$

as desired. ◀

We will now state the key lemma for this section. Namely, that π^s selects the i-th revealed active element with probability independent of the identities of the first i revealed elements.

▶ **Lemma 36.** *Against adversaries in $\mathcal{H}$, the probability that π^s selects the i-th revealed element, conditioned on it being active, is given by a function $g(i)$ that is independent of the identities of the first i revealed elements and their order.*

Proof. Consider any adversary $\mathcal{A} \in \mathcal{H}$. By definition $\mathcal{A}$ decides on the order in which to reveals the elements apriori, and reveals all active elements before all inactive ones. Suppose that the first i elements that $\mathcal{A}$ reveals are $a_1, \ldots, a_i$ in that order. To prove the lemma it is enough to show that

$$\Pr(\pi^s \text{ selects } a_i | a_1, \ldots, a_i \text{ are revealed}) = g(i)$$

where g is allowed to depend only on π. Note that the permutation σ drawn is independent of the order of the elements $a_1, \ldots, a_i$ by definition of $\mathcal{H}$. Thus, we have

$$\Pr(\pi^s \text{ selects } a_i | a_1, \ldots, a_i \text{ are revealed}) =$$

$$= \sum_\sigma \Pr(\pi_\sigma \text{ selects } a_i | a_1, \ldots, a_i \text{ are revealed}) \Pr(\sigma \text{ is drawn} | a_1, \ldots, a_i \text{ are revealed})$$

$$= \sum_\sigma \Pr(\pi_\sigma \text{ selects } a_i | a_1, \ldots, a_i \text{ are revealed}) \frac{1}{n!}$$

$$= \frac{1}{n!} \sum_\sigma \Pr(\pi_\sigma \text{ selects } a_i | a_1, \ldots, a_i \text{ are revealed})$$

$$= \frac{1}{n!} \sum_\sigma \sum_{B \subseteq \{a_1, \ldots, a_{i-1}\}} \Pr(\pi_\sigma \text{ selects } a_i \text{ and } \pi_\sigma \text{ selected } B | a_1, \ldots, a_i \text{ are revealed})$$

$$= \frac{1}{n!} \sum_\sigma \sum_{B \subseteq \{a_1, \ldots, a_{i-1}\}} \Pr(\pi \text{ sel. } \sigma^{-1}(a_i) \text{ and } \pi \text{ sel. } \sigma^{-1}(B) | \sigma^{-1}(\{a_1, \ldots, a_i\}) \text{ revealed})$$

In the second inequality we used the fact that the event that σ is drawn is independent of the decision of the adversary for which $a_1, \ldots, a_i$ to reveal. In the last equality we used Definition 28. Let's denote by (Δ) the last term above.

The key observation is that expression (Δ) does not depend on the elements $a_1, \ldots, a_i$ but only on i. Notice that for fixed A', B', a', such that $|A'| = i$, $B' \subset A'$ and $a' \in A' \setminus B'$, there are exactly

$$\binom{i-1}{|B'|}(|B'|)!(i-1-|B'|)!(n-i)!$$

terms in the sum (Δ) of the form

$$\Pr(\pi \text{ selects } a' \text{ and } \pi \text{ selected } B' | A' \text{ revealed})$$

To see this consider the number of (σ, B) which are solutions to

$$\sigma^{-1}(a_i) = a', \sigma^{-1}(B) = B', \sigma^{-1}(A) = A'$$

where $A = \{a_1, \ldots, a_i\}$. There are $\binom{i-1}{|B'|}$ ways to choose B. Given B there are $(|B'|)!(i-1-|B'|)!(n-i)!$ ways to choose σ in order to send a' to a_i, B' to B, and A' to A. Therefore, (Δ) is equal to

$$\sum_{|A'|=i, B' \subset A', a' \in A' \setminus B'} \Pr(\pi \text{ sel. } a' \text{ and } \pi \text{ sel. } B' | A' \text{ revealed}) \binom{i-1}{|B'|}(|B'|)!(i-1-|B'|)!(n-i)!$$

which only depends on i. Therefore, the probability that π^s selects the i-th revealed element given that it is active only depends on i we will denote it by $g(i)$. ◀

5.2 Upper bound on the probability of selection

In this section we will present an adversary in the class $\mathcal{H}$ and show how this adversary implies the upper bound of $1 - \Omega(\sqrt{\frac{\log(k)}{k}})$ on c. By Lemma 36 it follows that, against adversaries in $\mathcal{H}$, the probability that π^s selects the i-th revealed active element is given by a function $g(i)$.

We now describe the adversary. We will only specify what the adversary does when e_1 is active.

Adversary $\mathcal{A}^*$:

1. If there are m active elements except for e_1, the adversary computes $g(1), \ldots, g(m+1)$.
2. Before the process starts, the adversary finds $j = \arg\min_{i \in [1,m+1]} g(i)$ and reveals element e_1 at position j and all other m active elements on positions $j' \in [1, j-1] \cup [j+1, m+1]$ in arbitrary order.

It is not hard to see that the adversary $\mathcal{A}^*$ is in the class $\mathcal{H}$ because it commits to the order apriori and reveals all active elements before all inactive. We will next show the following lemma for the probability that π^s selects e_1 against the above adversary.

▶ **Lemma 37.** *Let $h(m+1) = \min_{i \in [1,m+1]} g(i)$ for $m \in [0, 2k-1]$. Then*

$$\Pr(\pi^s \text{ selects } e_1 \text{ against } \mathcal{A}^* | e_1 \text{ is active}) = \sum_{i=1}^{2k} h(i) \frac{\binom{2k-1}{i-1}}{2^{2k-1}}$$

Proof. Suppose e_1 is active. Notice that the number of active elements is equal to $N+1$, where $N \sim Bin(2k-1, \frac{1}{2})$. By definition of $\mathcal{A}^*$, when there are m active elements (except for e_1), the probability that π^s selects e_1 is equal to $h(m+1)$. Therefore, by the law of total probability, π^s selects e_1 with probability

$$\Pr(\pi^s \text{ selects } e_1 \text{ against } \mathcal{A}^* | e_1 \text{ is active}) = \sum_{i=1}^{2k} h(i) \Pr[N+1 = i]$$

$$= \sum_{i=1}^{2k} h(i) \Pr[Bin(2k-1, \frac{1}{2}) = i-1]$$

$$= \sum_{i=1}^{2k} h(i) \frac{\binom{2k-1}{i-1}}{2^{2k-1}}$$

as desired.   ◀

We will next show a property on the values $\{g(i)\}_{i=1}^{2k}$ that will be useful later.

▶ **Lemma 38.**

$$\sum_{i=1}^{2k} g(i) \leq k$$

Proof. Suppose that all $2k$ elements in N are active. We know that any adversary in $\mathcal{H}$ will choose an order for the elements to be revealed before π^s draws σ. In that case we know by Lemma 36 that π^s will select the i-th revealed element with probability $g(i)$. Let I_i be the indicator random variable that π^s selects that i-th revealed element. Note that $E[I_i] = g(i)$. Since π^s never selects more than k elements (Definition 1) we have that

$$k \geq \mathbb{E}[\sum_{i=1}^{2k} I_i] = \sum_{i=1}^{2k} \mathbb{E}[I_i] = \sum_{i=1}^{2k} g(i)$$

as desired.   ◀

As a last step towards Theorem 27, we consider a linear program relaxation whose optimal objective upper bounds the probability of selection c (Lemma 39), and characterize an optimal solution to the program (Lemma 40). Finally, we show that the optimal objective of the linear program is at most $1 - \Omega(\sqrt{\frac{\log(k)}{k}})$ (Lemma 41).

▶ **Lemma 39.** *Let c^* denote the optimal value of the following linear program*

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{2k} f(i) \frac{\binom{2k-1}{i-1}}{2^{2k-1}} \\
\text{subject to} \quad & \sum_{i=1}^{2k} f(i) \leq k \\
& f(i) \geq f(i+1) \ \text{for } i = 1, \ldots, 2k-1 \\
& f(2k) \geq 0
\end{aligned}
\tag{1}
$$

Then if π^s is c-selectable it holds that

$$c \leq c^*$$

Proof. We first claim that $\{h(i)\}_{i=1}^{2k}$ is a feasible assignment to (1). Notice that by definition we have that $h(i) \leq g(i)$ for $i \in [1, 2k]$. Using this combined with Lemma 38 we obtain

$$\sum_{i=1}^{2k} h(i) \leq \sum_{i=1}^{2k} g(i) \leq k$$

Further, note that by definition we have that $h(i) \geq h(i+1)$ for $i \in [1, 2k-1]$ and clearly $h(2k) \geq 0$. Combining the aforementioned observations we get that the vector $\{h(i)\}_{i=1}^{2k}$ is a feasible assignment of (1). Therefore, by Lemma 37 we obtain that

$$c^* \geq \sum_{i=1}^{2k} h(i) \frac{\binom{2k-1}{i-1}}{2^{2k-1}} = \Pr(\pi^s \text{ selects } e_1 \text{ against } \mathcal{A}^* | e_1 \text{ is active})$$

By Lemma 35 we know that π^s is c-selectable i.e.

$$\Pr(\pi^s \text{ selects } e_1 \text{ against } \mathcal{A}^* | e_1 \text{ is active}) \geq c$$

By combining the above two inequalities we obtain that

$$c^* \geq c$$

which finishes the proof. ◀

We will next prove a claim for the optimal solution of the linear program (1).

▶ **Lemma 40.** *The optimal solution of (1) has the form $f(i) = x$ for $i = 1 \ldots, k + a$, $f(k + a + 1) = y$, and $f(i) = 0$ for $i > k + a + 1$ for some $x \geq y \geq 0$ and $a \in [0, k]$.*

Proof. First, note that the constraints of the linear program (1) define a bounded convex polytope, so there exists a feasible assignment that achieves the optimum of (1). Let $b_i = \frac{\binom{2k-1}{i-1}}{2^{2k-1}}$, we know that $b_i < b_{i+1}$ for $i < k$, $b_k = b_{k+1}$, and $b_i > b_{i+1}$ for $i > k$. Let f^* be an optimal solution to (1). Suppose that $f^*(i) > f^*(i+1)$ for some $i < k$. Then consider f^{**} defined by $f^{**}(j) = f^*(j)$ for $j \notin \{i, i+1\}$, and $f^{**}(i) = f^{**}(i+1) = \frac{f^*(i)+f^*(i+1)}{2}$. Note that f^{**} still has decreasing non-negative entries and the sum of its entries is equal to that of f^* i.e. it is feasible. The difference between the objective values of f^{**} and f^* is equal to

$$\frac{f^*(i) + f^*(i+1)}{2}(b_i + b_{i+1}) - b_i f^*(i) - b_{i+1} f^*(i+1) = \frac{(f^*(i+1) - f^*(i))(b_i - b_{i+1})}{2} > 0$$

since $f^*(i+1) < f^*(i)$ and $b_i < b_{i+1}$. Thus, a contradiction with the optimality of f^*. Therefore, $f^*(i) = f^*(1)$ for all $i \leq k$.

Let $f^*(i) = x$ for $i \leq k$. If $f^*(j) \in \{x, 0\}$ for $j > k$ we are done. Otherwise let $j > k$ be the smallest index such that $x > f^*(j) > 0$, and let $f^*(j) = y$. This means $f^*(i) = x$ for $i < j$. Assume that $f^*(j + 1) > 0$. Then, let $l > j$ be the largest index such that $f^*(l) > 0$, and let $f^*(l) = z$. Choose $\epsilon < \min(x - y, z)$ and consider f^{**} defined by $f^{**}(j) = f^*(j) + \epsilon$, $f^{**}(l) = f^*(l) - \epsilon$, and $f^{**}(i) = f^*(i)$ for $i \notin \{j, l\}$. Note that f^{**} is feasible by the choice of ϵ since its entries are still decreasing and have the same sum as those of f. The difference between the objective of f^{**} and f^* equals to

$$(f^*(j) + \epsilon)b_j + (f^*(l) - \epsilon)b_l - f^*(j)b_j - f^*(l)b_l = \epsilon(b_j - b_l) > 0$$

as $b_j > b_l$ because $l > j > k$, which contradicts the optimality of f^*. Thus, we showed that $f^*(j + 1) = 0$, which finishes the proof of the Lemma. ◀

Note that by Lemma 40 we know that the optimal value of (1) has the following form

$$c^* = x \Pr(Bin(2k - 1, \tfrac{1}{2}) \leq k + a - 1) + y \Pr(Bin(2k - 1, \tfrac{1}{2}) = k + a) \tag{2}$$

for some $x \geq y \geq 0$ satisfying $x(k + a) + y \leq k$, where $a \geq 0$. By using these constraints we easily obtain that $y \leq x \leq \frac{k}{k+a}$. By using this inequality in equation (2), we obtain that

$$c^* \leq \frac{k}{k + a} \Pr(Bin(2k - 1, \tfrac{1}{2}) \leq k + a) \tag{3}$$

We now show the final Lemma of this section, which provides an upper bound for the RHS of (3). The proof of this lemma is purely technical so we refer the reader to the full version [10]

▶ **Lemma 41.** *Let $a \in [0, k]$, then*

$$\frac{k}{k + a} \Pr(Bin(2k - 1, \tfrac{1}{2}) \leq k + a) \leq 1 - \Omega\left(\sqrt{\frac{\log(k)}{k}}\right)$$

Proof. See full version [10] for a proof. ◀

▶ Remark 42. The optimal solution to the LP in Lemma 39 turns out not have the same values $f(i)$ as implied by the asymptotically optimal OCRS from [17] (Theorem 16). This is because the values implied by this OCRS would have an asymptotically optimal performance as opposed to exactly instance optimal.

5.3 Proof of Theorem 27

By Lemma 39, we know that

$$c \leq c^* \tag{4}$$

Additionally,by combining (3) and Lemma 41 we know that

$$c^* \leq 1 - \Omega\left(\sqrt{\frac{\log(k)}{k}}\right) \tag{5}$$

Combining (4) and (5) we obtain

$$c \leq 1 - \Omega\left(\sqrt{\frac{\log(k)}{k}}\right)$$

finishing the proof of Theorem 27.

6 Conclusion

We provide a new, simple, and optimal OCRS for k-uniform matroids against a fixed-order adversary. In particular, our algorithm has the advantage that it is extremely simple to implement and it does not require solving a mathematical program. Our analysis connects its performance to a random walk, and follows by concluding properties of this random walk. We expect that the tools we develop in analyzing our algorithm to be of independent interest and to have a broader applicability within online stochastic optimization.

As our second main result, we show that no OCRS for k-uniform matroids can be $(1 - \Omega(\sqrt{\frac{\log k}{k}}))$-selectable against an almighty adversary, establishing that the simple greedy OCRS implied by [17] is optimal.

───── **References** ─────

1 Marek Adamczyk and Michal Wlodarczyk. Random order contention resolution schemes. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 790–801, 2018. `doi:10.1109/FOCS.2018.00080`.

2 Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM Journal on Computing*, 43(2):930–972, 2014. `doi:10.1137/120878422`.

3 Nima Anari, Rad Niazadeh, Amin Saberi, and Ali Shameli. Nearly optimal pricing algorithms for production constrained and laminar bayesian selection. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019.*, pages 91–92, 2019. `doi:10.1145/3328526.3329652`.

4 Pablo D Azar, Robert Kleinberg, and S Matthew Weinberg. Prophet inequalities with limited information. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1358–1377. SIAM, 2014.

5 Constantine Caramanis, Paul Dütting, Matthew Faw, Federico Fusco, Philip Lazos, Stefano Leonardi, Orestis Papadigenopoulos, Emmanouil Pountourakis, and Rebecca Reiffenhäuser. Single-sample prophet inequalities via greedy-ordered selection. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1298–1325, 2022. `doi:10.1137/1.9781611977073.54`.

6 Shuchi Chawla, Nikhil Devanur, and Thodoris Lykouris. Static pricing for multi-unit prophet inequalities. *arXiv preprint arXiv:2007.07990*, 2020.

7 Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-Parameter Mechanism Design and Sequential Posted Pricing. In *the 42nd ACM Symposium on Theory of Computing (STOC)*, 2010.

8 Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.

9 José R. Correa, Andrés Cristi, Andrés Fielbaum, Tristan Pollner, and S. Matthew Weinberg. Optimal item pricing in online combinatorial auctions. In Karen Aardal and Laura Sanità, editors, *Integer Programming and Combinatorial Optimization - 23rd International Conference, IPCO 2022, Eindhoven, The Netherlands, June 27-29, 2022, Proceedings*, volume 13265 of *Lecture Notes in Computer Science*, pages 126–139. Springer, 2022. `doi:10.1007/978-3-031-06901-7_10`.

10 Atanas Dinev and S Matthew Weinberg. Simple and optimal online contention resolution schemes for k-uniform matroids. *arXiv preprint arXiv:2309.10078*, 2023.

11 Paul Duetting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet inequalities made easy: Stochastic optimization by pricing non-stochastic inputs. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 540–551, 2017. `doi:10.1109/FOCS.2017.56`.

12 Paul Dütting and Robert Kleinberg. Polymatroid prophet inequalities. In Nikhil Bansal and Irene Finocchi, editors, *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *Lecture Notes in Computer Science*, pages 437–449. Springer, 2015. `doi:10.1007/978-3-662-48350-3_37`.

13 Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. Online stochastic max-weight matching: Prophet inequality for vertex and edge arrival models. In Péter Biró, Jason D. Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, pages 769–787. ACM, 2020. `doi:10.1145/3391403.3399513`.

14 Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1014–1033. SIAM, 2016.

15 Oliver Göbel, Martin Hoefer, Thomas Kesselheim, Thomas Schleiden, and Berthold Vöcking. Online independent set beyond the worst-case: Secretaries, prophets, and periods. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, pages 508–519, 2014. `doi:10.1007/978-3-662-43951-7_43`.

16 Nikolai Gravin and Hongao Wang. Prophet inequality for bipartite matching: Merits of being simple and non adaptive. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019.*, pages 93–109, 2019. `doi:10.1145/3328526.3329604`.

17 Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI*, volume 7, pages 58–65, 2007.

18 Jiashuo Jiang, Will Ma, and Jiawei Zhang. Tight guarantees for multi-unit prophet inequalities and online stochastic knapsack. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1221–1246. SIAM, 2022.

19 Jiashuo Jiang, Will Ma, and Jiawei Zhang. Tightness without counterexamples: A new approach and new results for prophet inequalities. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC '23, page 909, New York, NY, USA, 2023. Association for Computing Machinery. `doi:10.1145/3580507.3597734`.

20 Danish Kashaev and Richard Santiago. A simple optimal contention resolution scheme for uniform matroids. *Theoretical Computer Science*, 940:81–96, 2023.

21 Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 123–136, 2012. `doi:10.1145/2213977.2213991`.

22 Ulrich Krengel and Louis Sucheston. On semiamarts, amarts, and processes with finite value. *Advances in Probability and Related Topics*, 4:197–266, 1978.

23 Euiwoong Lee and Sahil Singla. Optimal Online Contention Resolution Schemes via Ex-Ante Prophet Inequalities. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 57:1–57:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.ESA.2018.57`.

24 Calum MacRury, Will Ma, and Nathaniel Grammel. On (random-order) online contention resolution schemes for the matching polytope of (bipartite) graphs. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1995–2014. SIAM, 2023.

25 Aviad Rubinstein. Beyond matroids: secretary problem and prophet inequality with general constraints. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 324–332, 2016. `doi:10.1145/2897518.2897540`.

26 Aviad Rubinstein and Sahil Singla. Combinatorial prophet inequalities. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1671–1687, 2017. `doi:10.1137/1.9781611974782.110`.

27 Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *Annals of Probability*, 12(4):1213–1216, 1984.

28 Richard Santiago, Ivan Sergeev, and Rico Zenklusen. Simple random order contention resolution for graphic matroids with almost no prior information. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 84–95. SIAM, 2023.