Rényi-Ulam Games and Online Computation with Imperfect Advice

Spyros Angelopoulos
CNRS and LIP6-Sorbonne University, Paris, France

Shahin Kamali
York University, Toronto, Canada

Abstract

We study the nascent setting of online computation with *imperfect* advice, in which the online algorithm is enhanced by some prediction encoded in the form of an imperfect, and possibly erroneous binary string. The algorithm is oblivious to the advice error, but defines a desired *tolerance*, namely an upper bound on the number of erroneous advice bits it can tolerate. This is a model that generalizes the *Pareto-based* advice model, in which the performance of the algorithm is only evaluated at the extreme values of error (namely, if the advice has either no errors, or if it is generated adversarially). It also subsumes the model in which the algorithm elicits a prediction on the online sequence, via imperfect responses to a number of *binary queries*.

In this work, we establish connections between games with a lying responder, also known as *Rényi-Ulam games*, and the design and analysis of online algorithms with imperfect advice. Specifically, we demonstrate how to obtain upper and lower bounds on the competitive ratio for important online problems such as time-series search, online bidding, and fractional knapsack. Our techniques provide the first lower bounds for online problems in this model. We also highlight and exploit connections between competitive analysis with imperfect advice and fault-tolerance in multiprocessor systems. Last, we show how to waive the dependence on the tolerance parameter, by means of resource augmentation and robustification.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Online computation, Rényi-Ulam games, query models, beyond worst-case analysis

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.13

Related Version *Full Version*: `arxiv.org/abs/2301.01631`

Funding The first author was partially funded by the grant ANR-19-CE48-0016 from the French National Research Agency (ANR).The second author was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference number DGECR-2018-00059].

1 Introduction

Online computation, and *competitive analysis*, in particular, have served as the definitive framework for the theoretical analysis of algorithms in a state of uncertainty. While the early, standard definition of online computation [37] assumes that the algorithm has no knowledge in regard to the request sequence, in practical situations, the algorithm may indeed have certain limited, but possibly inaccurate such information (e.g., some lookahead, or historical information on typical sequences). Hence, there is a clear need for more nuanced models that capture the power and limitations of online algorithms enhanced with external information.

One such approach, within Theoretical Computer Science, is the framework of *advice complexity*; see [18, 9, 20], the survey [10] and the book [25]. In the advice-complexity model (and in particular, the *tape* model [8, 9]), the online algorithm receives a string that encodes information concerning the request sequence, and which can help improve its performance. The objective is to quantify the tradeoffs between the size of the advice (in terms of the number of bits) and the competitive ratio of the algorithm. This model places stringent requirements: the advice is assumed to be error-free, and may be provided by an omnipotent oracle. Thus, as noted in [34], this model is mostly of theoretical significance.

A different and more practical approach studies the effect of *predictions* towards improving the competitive ratio. In this model, the online algorithm is enhanced with some imperfect information concerning the request sequence, without restrictions on its size. One is interested in algorithms whose performance degrades gently as a function of the prediction *error*, and specifically perform well if the prediction is error-free (what is called the *consistency* of the algorithm), but also remain robust under any possible error (what is called the *robustness* of the algorithm). This line of research was initiated with the works [31] and [35], and a large number of online problems have been studied under this model (see, e.g., the survey [34] and the online collection [29]).

A combination of the advice complexity and prediction models is the *untrusted or Pareto-based advice* model, introduced in [5]. Here, parts of the advice may be erroneous, and the algorithm's performance is evaluated in two extreme situations, in regard to the advice error. At the one extreme, the advice is error-free, whereas, at the other extreme, the advice is generated by a (malicious) adversary who aims to maximize the performance degradation of the algorithm. Using the terminology of algorithms with predictions, these two competitive ratios are called consistency and robustness, respectively. The objective is to identify algorithms that are *Pareto-efficient*, and ideally *Pareto-optimal*, i.e., attain the best-possible tradeoffs between these two extreme measures. Several online problems have been studied recently within this framework of Pareto-optimality (both within the advice and the predictions models); see, e.g., [39, 28, 26, 4, 6].

1.1 Online computation with imperfect advice

The starting observation that motivates this work is that the Pareto-based framework of untrusted advice only focuses on extreme competitive ratios, namely the consistency and the robustness. A more general issue, instead, is to evaluate the impact of the advice error on the performance of the online algorithm. Given an advice string of size k, let us denote by $\eta \leq k$ the number of erroneous bits. Naturally, the algorithm does not know the exact advice error ahead of time. Instead, the algorithm defines an application-specific parameter $H \leq k$ which determines the desired *tolerance* to errors, or, equivalently, an anticipated upper bound on the advice error. This is motivated by recent works in learning-enhanced online algorithms with *weak predictions*, in which the prediction is an upper bound of some pertinent parameter of the input (see e.g., online knapsack with frequency predictions [23], where the prediction is an upper bound on the number of items of each value that appear online). Our objective is to quantify the tradeoffs between advice size, tolerance and competitive ratio, both from the point of upper and lower bounds.

A different interpretation of imperfect advice treats each advice bit as a (potentially erroneous) response to a *binary query* concerning the input. Hence, one may think of k-bit advice as a prediction elicited by means of k imperfect binary experts. Note that queries are known to help improve the performance of approximation algorithms in ML applications. For example, [33] studied clustering with noisy queries, where a query asks whether two points should belong in the same cluster, and where each query receives a correct response with probability p that is known to the algorithm. A different example is parsimonious learning-augmented caching [22], in which the system learns the predicted next-arrival time of certain appropriately queried pages.

In this work, we study the power, but also the limitations of online algorithms with adversarially erroneous queries. Unlike [33], we do not rely on any probabilistic assumptions concerning the query responses. To our knowledge, the imperfect advice model (in particular, its binary query-based interpretation) has only been applied to the problems of *contract scheduling* [6] and time-series search [7], from the point of view of upper bounds. While these works showed that binary queries help improve the algorithmic performance, both in terms of theoretical and empirical analysis, no principled methodology for obtaining lower bounds has been developed so far.

1.2 Contribution

We establish connections between games with a lying responder and the design and analysis of online algorithms with imperfect advice. Namely, we show how to leverage results from the analysis of *Rényi-Ulam* games, and obtain both positive and negative results on the competitive analysis. We apply these tools to three important and well-studied online problems, namely time-series search, online bidding, and online fractional knapsack. Our results improve the known upper bounds for these problems, where such results were already known, but also provide the first lower bounds on the competitive ratio of online problems in this setting, without any restrictive assumptions.

More precisely, we begin as a warm-up[1] with the time-series search problem in Section 3, which illustrates how these techniques can help us improve upon the results of [7]; we also show how to evaluate the competitive ratios, using approximations based on the binary entropy function. In Section 4, we study a more complex application, namely the online bidding problem, first studied in [5] in the context of untrusted advice. Here, the crucial part is establishing near-optimal lower bounds. We achieve this by formulating a multi-processor version of online bidding in $l \leq 2^k$ processors, in which a certain number of processors may be faulty; we then relate the competitive ratio of this problem to the imperfect advice setting, by relating fault-tolerance in the processor level, to the inherent error in Rényi-Ulam games. In Section 5 we study the online fractional knapsack problem. Here, we present an algorithm whose competitive ratio converges to 1 at a rate exponential in k, as long as $H < k/2$. We also present a near-matching lower bound that shows that our algorithm is close-to-optimal. For the upper bound, the crux is to allocate queries so as to approximate two appropriately defined parameters of the instance. For the lower bound, we use an information theoretic argument. Specifically, we show a reduction from Rényi-Ulam games: if there existed an algorithm of competitive ratio better than a certain value, one could play the game beyond the theoretical performance bound, which is a contradiction.

As explained above, the parameter H expresses the algorithm's desired tolerance to errors, and is thus application-specific. In Section 6 we show how to waive the assumption that the precise tolerance is known ahead of time, in two different ways: First, by *resource-augmentation* arguments, i.e., by comparing the performance of an algorithm with perfect (error-free) advice of size k to that of an algorithm with $l > k$ advice bits but potentially very high advice error. Second, by *robustifying* the algorithm, namely by requiring that the algorithm performs well even if the error happens to exceed the tolerance parameter.

The techniques we develop can be applicable to other online problems. Specifically, our approach to the online bidding problem defines the following general framework: For upper bounds, one would aim to define a collection of "candidate" algorithms that are closely

[1] For *ski rental*, which is another canonical warm-up problem, [5] showed that a single advice bit suffices to obtain optimal consistency/robustness. Hence, the problem is resolved under the imperfect advice model as well.

ranked in terms of their worst-case performance. Then the advice can be used so as to select a suitable candidate from this collection that is close to the best-possible. For lower bounds, one would aim to show that in any collection of candidate algorithms, the erroneous queries may have to always return a solution sufficiently far, in terms of "rank", from the best one; then one needs to relate the concept of "rank" to performance, from a lower-bound point of view. This last part highlights connections between an online problem with adversarial advice and its fault-tolerant version in a parallel system (with no advice). On the other hand, our approach to the time-series and fractional knapsack problems illustrate another general technique: For upper bounds, one should identify some important parameters of the problem, then allocate the queries appropriately so as to approximate them in the presence of response errors. For lower bounds, information-theoretic arguments should establish a reduction from a Rényi-Ulam game to the online problem.

There are two additional observations concerning the results in this work. First, we allow *adaptive* queries, in that the response to the i-th query is a function of responses to the previous $i-1$ queries. Second, it is important to note that the results we present cannot be obtained straightforwardly by applying some error-correcting code. More precisely, one may be tempted to dedicate some advice bits towards error correction and use the remaining error-free bits in the spirit of classic advice complexity results. However, such an approach may very well be suboptimal since, depending on the problem at hand, an optimal algorithm may benefit more from a large number of somewhat erroneous advice bits than from a smaller number of perfect bits, and the analysis must take into account this possibility.

Due to space limitations, we omit or only sketch certain technical proofs. We refer to the full version on arXiv for the complete proofs.

2 Games with a lying responder

We review some core results related to games with a lying responder which will be in the heart of the analysis of online problems with imperfect advice. We are particularly interested in [36], which studied games between a *questioner* and a *responder*, related to an unknown value x drawn from a domain $\mathcal{D}$, where D is a subset of reals or in general a totally ordered set. The questioner may ask general queries of the form "is x in S", where S is some subset of $\mathcal{D}$, and which are called *subset* queries. The upper bounds of [36] hold even if the questioner asks much simpler queries, namely *comparison* queries of the form "is x at most a", for some given a. Both the upper and lower bounds in [36] are expressed in terms of partial sums of binomial coefficients. Formally, we define:

$$\left(\!\!\binom{N}{m}\!\!\right) := \sum_{j=0}^{m} \binom{N}{j}, \text{ for } m \le N.$$

We are interested, in particular, in the following game played over a continuous space:

ContinuousSearch(k, H) game. In this game, x is a real number with $x \in \mathcal{D} = (0,1]$, and the questioner asks k queries, at most H of which may receive erroneous responses. The objective of the questioner is to find an interval I_x such that $x \in I_x$ and $|I_x|$ is minimized.

▶ **Lemma 1** ([36]). *Any questioner's strategy for* ContinuousSearch(k, H) *with* $H \le k/2$ *is such that* $|I_x| \ge \left(\!\binom{k}{H}\!\right)/2^k$. *Moreover, for* $H \le k/2$, *there is a strategy, named* C-Weighting, *that uses comparison queries and outputs an interval* $I_{W,x}$ *with* $|I_{W,x}| \le \left(\!\binom{k-H}{H}\!\right)/2^{k-H}$.

The following game will be useful in our analysis of online time-series and fractional knapsack.

FIND(k, H) game. In this game, given k and $H \leq k/2$, and $\mathcal{D} = \{1, \ldots, m\}$, the objective is to find an unknown $x \in \mathcal{D}$, using k queries, up to H of which may be answered incorrectly.

The proof of the following theorem is direct from Lemma 1:

▶ **Theorem 2.** *The largest positive integer $\mu(k, H)$ such that a questioner can identify any number $x \in \{1, 2, \ldots, \mu(k, H)\}$ in the* Find(k, H) *game is such that*

$$2^{k-H} / \left(\!\!\binom{k-H}{H}\!\!\right) \leq \mu(k, H) \leq 2^k / \left(\!\!\binom{k}{H}\!\!\right).$$

We define two further games that will be of interest to our analysis. The first is related to searching in cyclic permutations, and will be useful in the upper-bound analysis of online bidding.

MINCYCLIC(n, k, H) game. Given an array $A[0 \ldots n-1]$ whose elements are an unknown cyclic permutation of $\{0, \ldots, n-1\}$, the objective is to use k queries, at most $H \leq k/2$ of which can be erroneous, so as to output an index of the array whose element is as small as possible.

▶ **Theorem 3.** *There is a questioner's strategy for* MinCyclic(n, k, H) *based on k comparison queries that outputs an index j such that $A[j] \leq \lceil n \left(\!\binom{k-H}{H}\!\right) / 2^{k-H} \rceil$, for all $H \leq k/2$.*

Last, we define a game that is related to searching in general permutations, and it will be useful in establishing lower bounds on the competitiveness of online bidding.

SEARCH(n, k, H) game. Given an array, $A[0, \ldots, n-1]$ whose elements are an unknown permutation of $\{0, \ldots, n-1\}$, the objective is to use k queries, at most H of which can be erroneous, so as to output an index of the array whose element is as small as possible.

▶ **Theorem 4.** *For any questioner's strategy for the* Search(n, k, H) *game, there is a responder's strategy such that if e is the element of A that is returned, then $A[e] \geq \lfloor n \left(\!\binom{k}{H}\!\right) / 2^k \rfloor$.*

3 A warm-up: Online time-series search

The *online (time series) search* problem formulates a simple, yet fundamental setting in decision-making under uncertainty. In this problem, a player must sell an indivisible asset within a certain time horizon, e.g., within a certain number of days d, that is unknown to the player. On each day i, a *price* p_i is revealed, and the player has two choices: either accept the price, and gain a profit p_i (at which point the game ends), or reject the price (at which point the game continues to day $i+1$). If the player has not accepted a price by day d, then it accepts by default the last price p_d. The competitive ratio of the player's algorithm is the worst-case ratio, over all price sequences, of the maximum price in the sequence divided by the price accepted by the player.

The problem was introduced and studied in [19] that gave a simple, deterministic algorithm that achieves a competitive ratio equal to $\sqrt{M/m}$, where M, m are upper and lower bounds on the maximum and minimum price in the sequence, respectively, and which are assumed to be known to the algorithm. This bound is optimal for deterministic algorithms. Time-series search is a basic paradigm in online financial optimization, and several variants and generalizations have been studied [17, 30, 40, 16]; see also the survey [27]. The problem has also been used as a case study for evaluating several performance measures of online algorithms, including measures alternative to competitive analysis [11, 1].

Time-series search was recently studied under the imperfect advice framework in [7], who showed an upper bound of $(M/m)^{2^{2H-k/2}}$ on the competitive ratio with k-bit advice and tolerance H, under the assumption that $H \leq k/4$. Note that no upper bound is known for $H \in (k/4, k/2]$. If the advice is error-free, i.e., in the advice-complexity model, then a tight bound on the competitive ratio equal to $(M/m)^{\frac{1}{2^k+1}}$ is due to [16].

We show the following result, as an application of the $\textsc{Find}(k, H)$ game discussed in Section 2.

▶ **Theorem 5.** *Consider the online time series search problem, with imperfect advice of size k and tolerance $H \leq k/2$. There is an algorithm that uses k comparison queries, and that has competitive ratio at most $(M/m)^{\frac{1}{U+1}}$, where $U = \lfloor 2^{k-H} / \left(\!\binom{k-H}{H}\!\right) \rfloor$, for any $H \leq k/2$. In contrast, no (deterministic) algorithm based on k subset queries has competitive ratio less than $(M/m)^{\frac{1}{L+1}}$, where $L = \lceil 2^k / \left(\!\binom{k}{H}\!\right) \rceil$.*

Proof. We first show the upper bound. Let $a_1, \ldots a_U, r$ be defined such that $r = \frac{a_1}{m} = \frac{a_2}{a_1} = \ldots = \frac{a_U}{a_{U-1}} = \frac{M}{a_U}$, hence $r = (M/m)^{1/(U+1)}$. The algorithm uses k comparison queries so as to find the best *reservation* price, in the set $\{a_i\}_{i=1}^U$, i.e., a threshold p above which the algorithm will always accept a price in the sequence. In particular, it can choose p to be the maximum value in $\{a_i\}_{i=1}^U$ that does not exceed the maximum price in the sequence. This follows from Theorem 2, since $U \leq 2^{k-H} / \left(\!\binom{k-H}{H}\!\right)$. From the definition of the set $\{a_i\}_{i=1}^U$, it easily follows that this algorithm has competitive ratio at most r, which completes the proof of the upper bound.

We now show the lower bound. By way of contradiction, suppose that there is an algorithm A for time-series search with k-bit imperfect advice, and of competitive ratio less than $C = (M/m)^{\frac{1}{L+1}}$. We will show that A could then be used in the $\textsc{Find}(k, H)$ game so as to identify, using k queries, an unknown value in $\{1, \ldots, L+1\}$, which is a contradiction to the upper bound of Theorem 2.

To arrive at the contradiction, define $a_1, \ldots, a_L$ and r' such that

$$r' = \frac{a_1}{m} = \frac{a_2}{a_1} = \ldots = \frac{a_L}{a_{L-1}} = \frac{M}{a_L},$$

hence $r' = (M/m)^{\frac{1}{L+1}} = C$. Consider a game between the online algorithm A and the adversary, in which the request sequences consist of prices in $\{m, a_1, \ldots, a_L, M\}$. More precisely, consider the set of request sequences of the form $\sigma_i = m, a_1, \ldots, a_i$, for all $i \in [1, L+1]$, where a_{L+1} is defined to be equal to M. In σ_i, A must accept price a_i to be strictly less than C-competitive. Equivalently, A uses k queries with at most H errors, and finds a_i in the set $\{a_j\}_{j=1}^{L+1}$, which contradicts Theorem 2. ◀

3.1 Comparison of the bounds

In order to compare the upper and lower bounds of Theorem 5, we need to be able to evaluate the partial sum of binomial coefficients. Since this partial sum does not have a closed form, we will rely on the following useful approximation from [32]. Let $\mathcal{H}$ denote the *binary entropy* function. Then

$$\frac{2^{N\mathcal{H}(\frac{m}{N})}}{\sqrt{8m(1-\frac{m}{N})}} \leq \left(\!\binom{N}{m}\!\right) \leq 2^{N\mathcal{H}(\frac{m}{N})}, \quad \text{for } 0 < m < N/2. \tag{1}$$

We will also use the following property of the binary entropy function

$$4p(1-p) \leq \mathcal{H}(p) \leq (4p(1-p))^{1/\ln 4}, \quad \text{for all } p \in (0,1). \tag{2}$$

We first show that the algorithm of Theorem 5 improves upon the one of [7]. First, note that [7] assumes that $H \leq k/4$, whereas Theorem 5 applies to all $H \leq k/2$. Furthermore, we improve on the competitive ratio for all values of H and k. For this, it suffices to show that $\left(\!\binom{k-H}{H}\!\right)/2^{k-H} < 2^{2H-k/2}$, which, from (1) holds if $2^{(k-H)(\mathcal{H}(\frac{H}{k-H})-1)} < 2^{2H-k/2}$, or equivalently $(k-H)(\mathcal{H}(\frac{H}{k-H})-1) < 2H-k/2$. Let τ be such that $\tau = H/k$ (hence $\tau \leq 1/2$), then the latter is equivalent to showing that $\mathcal{H}(\frac{\tau}{1-\tau}) < \frac{1+2\tau}{2-2\tau}$. Using (2), it suffices to show that

$$(\frac{4\tau(1-2\tau)}{(1-\tau)^2})^{1/\ln 4} < \frac{1+2\tau}{2-2\tau},$$

which holds for all $\tau \leq 1/2$.

Next, we investigate how close the upper and lower bounds of Theorem 5 are to each other. Recall that the bounds are of the form $(M/m)^{1/(U+1)}$, and $(M/m)^{1/(L+1)}$. Using (1), and ignoring for simplicity the floors and ceilings, we obtain that

$$U \geq 2^{k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))} \text{ and } L \leq \sqrt{8k\tau(1-\tau)}2^{k(1-\mathcal{H}(\tau))}.$$

The above inequalities, along with (2) show that the upper and lower bounds are very close to each other, since for any fixed value of τ, we have that $U \geq 2^{\Theta(k)}$ and $L \leq 2^{\Theta(k)}$.

4 Online bidding

Online bidding was introduced in [15] as a canonical problem for formalizing doubling-based strategies in online and offline optimization problems, such as searching for a target on the line, minimum latency, and hierarchical clustering. In this problem, a player wants to guess a hidden, unknown real value $u \geq 1$. To this end, the player defines an (infinite) sequence $X = (x_i)$ of positive, increasing *bids*, which is called its *strategy*. The *cost of discovering* the hidden value u using the strategy X, denoted by $c(X, u)$, is defined to be equal to $\sum_{i=1}^{j_u} x_i$, where j_u is such that $x_{j_u-1} < u \leq x_{j_u}$. Hence one naturally defines the competitive ratio of the bidder's strategy X as $\mathrm{Cr}(X) = \sup_u \frac{c(X,u)}{u}$.

In the standard version of the problem, i.e, assuming no advice, the doubling strategy $x_i = 2^i$ achieves optimal competitive ratio equal to 4. Online bidding was studied under the untrusted advice model in [5], which gave bounds on the consistency/robustness tradeoffs. It was also studied under a model in which the prediction is the hidden value in [3, 5]. The problem is related to contract scheduling, studied in [6], see also the discussion in Section 4.1.3.

4.1 Online bidding with imperfect advice

4.1.1 Upper bound

The idea behind the upper bound is as follows. We will consider bidding sequences from a space of 2^k geometrically-increasing sequences (see Definition 6). In the ideal situation of perfect advice, the k advice bits could be used to identify the best strategy in this space. In the presence of advice errors, we will show how to exploit the cyclic structure of this space, in conjunction with our upper bound for the MinCyclic game (Theorem 3), so as to find a strategy that is not too far from the optimal.

We first define the space of geometrically-increasing bidding sequences.

▶ **Definition 6.** *For given $b > 1$, and $l \in \mathbb{N}^+$ define $\mathcal{X}_{b,l}$ as the set of bidding sequences $\{X_0, \dots X_{l-1}\}$, in which $X_i = (b^{i+jl})_{j=0}^{\infty}$, for all $i \in [0, l-1]$.*

From the definition of $\mathcal{X}_{b,l}$, it is easy to see that for any potential target u, there is a cyclic permutation π of $\{0, \dots l-1\}$ which determines an ordering of the strategies in $\mathcal{X}_{b,l}$ in terms of their performance. More precisely, suppose that $X_{\pi(0)}$ is the best sequence that discovers u at least cost, say C. Then $X_{\pi(i)}$ discovers u at cost at most $b^i C$. This property can help us show the following upper bound:

▶ **Theorem 7.** *There is a bidding strategy based on k comparison queries of competitive ratio at most $\frac{1+U}{2^k}\left(1+\frac{2^k}{1+U}\right)^{1+\frac{1+U}{2^k}}$, where $U = \lceil 2^H \left(\!\binom{k-H}{H}\!\right) \rceil$.*

4.1.2 Lower bound

The idea behind the lower bound is as follows. With k advice bits, the best one can do is choose the best strategy from a set $\mathcal{X}$ that consists of at most 2^k strategies. Note that if the advice were error-free, $|\mathcal{X}|$ could be as large as 2^k; however, in the presence of errors, the algorithm may choose to narrow $|\mathcal{X}|$.

Our approach combines two ideas. The first idea uses the abstraction of the Search(n, k, H) game, and the lower bound of Theorem 4. This result will allow us to place a lower bound on the rank of the chosen strategy, where the best strategy has rank 0. The second idea is to define a *measure* that relates how much worse a strategy of rank j in $\mathcal{X}$ has to be relative to the best strategy in $\mathcal{X}$. We will accomplish this by appealing to the concepts of *parallelism* and *fault tolerance*.

More precisely, given integers p, and ϕ, with $\phi < p$, we define the *fault-tolerant parallel bidding* problem, denoted by $\texttt{FPB}(p, \phi)$, as follows. The player is allowed to run, in parallel, p bidding strategies; however, ϕ of these strategies can be *faulty*, in that they never discover the target; e.g., we can think of a fault strategy as one in which the player abruptly stops submitting bids, at some point in time, akin to a "byzantine" failure. The cost of discovering a target u is then defined as the minimum cost at which one of the $p - \phi$ non-faulty strategies discovers the target, noting that the faults are dictated by an adversary that aims to maximize this cost. The competitive ratio is defined accordingly.

The next theorem is the main technical result for $\texttt{FPB}(p, \phi)$, which gives a lower bound on the competitive ratio of any strategy for this problem, as a function of the parameters p, ϕ and $\alpha_{\bar{X}}$. Here, $\bar{X}$ is defined as the sorted sequence of all bids in the p-parallel strategy X, in non-decreasing order. Moreover, given a sequence X of positive reals, we define α_X to be equal to $\limsup_{i\to\infty} x_i^{1/i}$.

▶ **Theorem 8.** *Every p-parallel strategy X for $\texttt{FPB}(p, \phi)$ has competitive ratio $Cr(X) \geq \frac{\alpha_{\bar{X}}^{p+1+\phi}}{\alpha_{\bar{X}}^{p}-1}$.*

Proof sketch. We use properties of p-parallel strategies so as to show that any such strategy satisfies $\mathrm{Cr}(X) \geq \sup_q \frac{\sum_{i=0}^{q+\phi+1} \bar{x}_i}{\sum_{i=q}^{q-(p-1)} \bar{x}_i}$. We then use Gal's functional theorem [21] to obtain the result. We omit several technical details. ◀

We now show how to obtain a lower bound for the problem by combining the above ideas. We emphasize a subtle point: unlike error-free advice of size k, where one should always choose the best strategy out of a collection of exactly 2^k strategies, it is conceivable that, in the presence of errors, this collection could very well be of size $l < 2^k$. This is because, as l decreases, so does the effect of errors on the competitive ratio. In other words, we need to establish the result for *all* values $l \leq 2^k$, and not only for $l = 2^k$.

▶ **Theorem 9.** *For every bidding sequence X and k subset queries in the imperfect advice model, we have $Cr(X) \geq \frac{1}{L}(1+L)^{1+1/L}$, where $L = 2^k / \left(\!\binom{k}{H}\!\right)$.*

Proof. Every bidding strategy will use the query responses so as to select a strategy from a set $\mathcal{X} = \{X_0, \ldots, X_{l-1}\}$ of candidate sequences, for some $l \leq 2^k$. For a given target value u, there is an ordering of the l sequences in $\mathcal{X}$ such that $X_{\pi(i)}$ has no worse competitive ratio than $X_{\pi(i+1)}$, namely the permutation orders the sequences in decreasing order of performance. From Theorem 4, it follows that the strategy will choose a sequence X_j such that $\pi(j) \geq \lfloor l \left(\!\binom{k}{H}\!\right) / 2^k \rfloor$. The competitive ratio of the selected sequence is at least the competitive ratio of the l-parallel strategy defined by $\mathcal{X}$, in which up to $\phi_l = \lfloor l \left(\!\binom{k}{H}\!\right) / 2^k \rfloor$ sequences may be faulty. From Theorem 8,

$$\mathrm{Cr}(X) \geq \frac{\alpha_{\bar{X}}^{l+1+\phi_l}}{\alpha_{\bar{X}}^{l} - 1}, \quad \text{with } \phi_l = \lfloor l \left(\!\!\binom{k}{H}\!\!\right) / 2^k \rfloor. \tag{3}$$

We now consider two cases. Suppose first that $l < L$. In this case, case $\phi_l = 0$, and therefore (3) implies that $\mathrm{Cr}(X) \geq \alpha_{\bar{X}}^{l+1}/(\alpha_{\bar{X}}^{l} - 1)$, which is minimized for $\alpha_{\bar{X}} = (l+1)^{1/l} > 1$, therefore $\mathrm{Cr}(X) \geq \frac{1}{l}(l+1)^{1+1/l}$. This function is decreasing in l, and since $l < L$ we have $\mathrm{Cr}(X) \geq \frac{1}{L}(1+L)^{1+1/L}$. Next, suppose that $l \in [L, 2^k]$. In this case, (3) gives $\mathrm{Cr}(X) \geq \frac{\alpha_{\bar{X}}^{l(1+1/L)}}{\alpha_{\bar{X}}^{l} - 1}$. The above expression is minimized for $\alpha_{\bar{X}} = (1+L)^{1/l}$, and by substitution we obtain again $\mathrm{Cr}(X) \geq \frac{1}{L}(1+L)^{1+1/L}$. ◀

4.1.3 Comparison of the bounds

We can prove that the ratio between the two bounds is approximately

$$\log \frac{\mathrm{UB}}{\mathrm{LB}} \leq \frac{\sqrt{8k\tau(1-\tau)}k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))}{2^{k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))}} - \frac{k(1-\mathcal{H}(\tau))}{2^{k(1-\mathcal{H}(\tau))}},$$

where $\tau = H/k$. We infer that as k increases, and for any fixed value of τ, the upper and lower bounds become very close to each other.

Note that the techniques of [6] imply an online bidding strategy with imperfect advice of competitive ratio roughly equal to $f(2k/H)$, where f is the decreasing function $f(x) = \frac{1}{x}(1+x)^{1+\frac{1}{x}}$. Thus, if $H = \Theta(k)$, then the competitive ratio is independent of the number of queries k. In contrast, the competitive ratio of Theorem 7 is roughly equal to $f(2^k/U)$, which is smaller than $f(2k/H)$, and which rapidly decreases as the number of queries k increases.

We also note that our analysis implies a tight bound on the advice complexity of online bidding. No previous bounds on the advice complexity of this problem were known.

5 Online fractional knapsack

In the online fractional knapsack problem, the request sequence consists of *items*, where item i has a *value* $v_i \in \mathbb{R}^+$ and a *size* $s_i \in (0, 1]$. The algorithm has a knapsack of unit capacity, and when considering item i, it can accept irrevocably a fraction $f_i \in (0, 1]$ of the item, subject to capacity constraints. More precisely, the algorithm aims to maximize $\sum_i (f_i \cdot v_i)$ subject to $\sum_i (f_i \cdot s_i) \leq 1$. Online fractional knapsack has important applications in sponsored search auctions, ad allocation and online trading, and has been studied in several settings, e.g., [2, 24, 38, 14]. In this section, we study this problem in the imperfect advice setting.

Let $d_i = v_i/s_i$ denote the *density* of item i. While the offline version of the problem admits an optimal solution via a simple greedy algorithm (that sorts all items by non-decreasing order of density, and accepts items in this order until the knapsack is full), the online version is more challenging. Suppose that $d_i \in [L, U]$, for L, U known to the algorithm. [13, 12] gave matching $O(\log(U/L))$ and $\Omega(\log(U/L))$ upper and lower bounds on the competitive ratio of the problem, respectively, and [41] showed an optimal bound of $\ln(U/L) + 1$ for deterministic algorithms.

5.1 Upper bound

As in all previous work, we assume that the density of all items is in $[L, U]$ for known values of L and U. Let d^* denote the smallest density of an item included at a positive fraction in the optimal solution. That is, the optimal algorithm Opt accepts a fraction 1 of items with density larger than d^*, and fills the remaining space with a fraction of items of density d^*. Unfortunately, knowing d^* (even its exact value) is not sufficient for an online algorithm to be anywhere as efficient as Opt. For example, an algorithm that accepts a fraction 1 of items of density larger than d^* has unbounded competitive ratio in sequences that consist only of items of density d^*. Similarly, an algorithm that accepts a fraction 1 of items with density at least d^* has unbounded competitive ratio in sequences in which items of density d^* appear early in the sequence, and items of greater density later in the sequence. However, if we denote by $c^* \in (0, 1)$ the fraction of the knapsack in the optimal solution that is either empty or occupied with items of density d^*, then knowing the exact value of both d^* and c^* suffices to achieve optimality. Our approach will then aim to use k comparison queries so as to approximate c^* and d^*, then use these approximations to choose fractional items.

5.1.1 Algorithm and analysis

We describe the online algorithm. We first define two types of partitions, related to the parameters d^* and c^*. In what concerns d^*, partition the interval $[L, U]$ into s sub-intervals $I_1, \ldots, I_s$ such that $I_i = [d_{i-1}, d_i)$, for s that will be specified later. We also set $L = d_0, U = d_s$. The values d_i are defined so that: $\beta = \frac{d_1}{d_0} = \frac{d_2}{d_1} = \ldots = \frac{d_s}{d_{s-1}}$. Thus, we have $\beta = (U/L)^{1/s}$ and $d_i = L \cdot \beta^i$, and note that $d^* \in I_x$ for some $x \in [1, s]$.

In what concerns the parameter c^*, we partition the interval $[0, 1]$ into m sub-intervals $I'_1, \ldots, I'_m$ such that $I'_i = [c_{i-1}, c_i)$; we have $c_0 = 0$ and $c_m = 1$. The value of m will be determined later; the values c_i are defined so that $c_1 = c_2 - \frac{c_1}{\beta} = c_3 - \frac{c_2}{\beta} = \ldots = c_m - \frac{c_{m-1}}{\beta}$.

It readily follows that for $i \geq 1$, we have $c_i = \frac{\beta^{m+i-1} - \beta^{m+i-2}}{\beta^m - 1}$. In particular, $c_1 = \frac{\beta^m - \beta^{m-1}}{\beta^m - 1}$, and $\frac{1}{1-c_1} = \frac{\beta^m - 1}{\beta^{m-1} - 1}$. Note also that $c^* \in I'_y$ for some $y \in [1, m]$.

Provided that $s \cdot m \leq \lfloor 2^{k-H} / \left(\!\binom{k-H}{H}\!\right) \rfloor$, Theorem 2 shows that the algorithm can use k comparison queries so as to identify both x and y. Given these values, the algorithm reserves, in its knapsack, a capacity $c = c_{y-1}$ for items with density in the range $I_x = [d_{x-1}, d_x)$, to which we refer as *critical items*. The algorithm uses the remaining capacity of $1 - c$ for items of density larger than d_x, to which we refer as *heavy items*, and accepts a fraction 1 of all critical items, as long as the capacity c reserved for them allows. Similarly, the algorithm accepts a fraction 1 of heavy items and places them in their dedicated space of the knapsack. Given that $c^* \in I_y$, we have $1 - c > 1 - c^*$; that is, the reserved capacity for heavy items is at least equal to the total size of these items. In other words, the algorithm can afford to accept all heavy items. The algorithm rejects all items of density smaller than d_{x-1}.

▶ **Theorem 10.** *For any $H \le k/2$, the above algorithm has competitive ratio*

$$\min_{s,m\in\mathbb{N}} \quad f_m(\beta) \qquad \text{where} \quad \beta = (U/L)^{1/s}, \quad \text{and} \quad f_m(\beta) = \frac{\beta^m - 1}{\beta^{m-1} - 1}$$

$$\text{subject to} \quad s \cdot m \le \lfloor 2^{k-H} / \left(\!\!\binom{k-H}{H}\!\!\right) \rfloor.$$

5.2 Lower bound

We will show a lower bound $C(k,H)$ on the competitive ratio of any algorithm with imperfect advice. For the sake of contradiction, suppose there is an algorithm A of competitive ratio better than $C(k,H)$. Our proof is based on a reduction from the Find(k,H) game. Specifically, we prove that, based on A, we obtain a questioner's strategy for Find(k,H) which can find a value $z \in \{1,\dots,p\}$, with $p = \lceil 2^k / \left(\!\binom{k}{H}\!\right) \rceil + 1$, which contradicts Theorem 2.

We give the intuition behind the proof. Let s and m be any two positive integers such that $s \cdot m \le p$ and $s \cdot (m+1) > p$. Define $\beta = (U/L)^{1/s}$, and $d_i = U \cdot \beta^i$, for $i \in [1,s]$. Given a pair (x,y) of integers, where $x \in \{1,\dots,s\}$ and $y \in \{1,\dots m+1\}$, define the sequence

$$\sigma_{x,y} = ((d_1,1),(d_2,1),\dots,(d_{x-1},1),(d_x,c_y)),$$

where (d_i,j) indicates a subsequence of j/ϵ items, each of which has size ϵ and density d_i, and where ϵ is infinitesimally small. $c_y \in [0,1]$ is defined appropriately in the proof. For this sequence, $\text{OPT}(\sigma_{x,y}) = (1-c_y)d_{x-1} + c_y d_x$. There are $s \cdot (m+1) > p$ such sequences, and $\sigma_{x,y}$ is a prefix sequence of $\sigma_{x,y+1}$, and $\sigma_{x,m}$ is a prefix sequence of $\sigma_{x+1,1}$. In the proof, we consider request sequences of this form, and we show that if A is $C(k,H)$-competitive, its decisions can help find any given $z \in \{1,\dots,p\}$, which contradicts Theorem 2.

▶ **Theorem 11.** *For the fractional knapsack problem, where items densities are in $[L,U]$, no deterministic algorithm with k subset queries, out of which $H \le k/2$ may have erroneous responses, can achieve a competitive ratio better than*

$$C(k,H) = \min_{s,m\in\mathbb{N}} \quad g_m(\beta) \quad \text{where} \quad \beta = (U/L)^{1/s}, \quad g_m(\beta) = \left(\frac{\beta^2 - \beta + 1}{2\beta + 1}\right)^{1/(m+1)}$$

$$\text{subject to} \quad s \cdot m \le \lceil 2^k / \left(\!\!\binom{k}{H}\!\!\right) \rceil + 1.$$

Comparison of the bounds

Let $\tau = H/k$. Since $\frac{\beta^m - 1}{\beta^{m-1}-1} \le \beta$, using (1), the upper bound of Theorem 10 is at most $(U/L)^q$, where $q \le 1/2^{k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))}$. Furthermore, since $\frac{\beta^2-\beta+1}{2\beta+1} \ge \frac{\beta}{3}$ (for all $\beta \ge 3$), the lower bound of Theorem 11 is at least $(U/L)^{q'}(1/3)^{q'}$, where $q' \ge 1/(2\sqrt{8k\tau(1-\tau)}2^{k(1-\mathcal{H}(\tau))}+1)$, for all $U/L \ge 3$. For simplicity, we omitted the floors and ceilings.

6 Waiving the assumption of the tolerance parameter

In the imperfect advice setting we have studied so far, the algorithm defines an application-specific tolerance parameter that measures its desired tolerance to errors (or equivalently, an anticipated upper bound on the error). This parameter is in a sense required, since the analysis of Rényi-Ulam games in [36] involves the extreme value of error (i.e., H) instead of the instance-specific error value (i.e., η). Nevertheless, in this section, we discuss how to mitigate the need for pre-determining a tolerance parameter. We propose two different approaches, based on *resource-augmentation*, and *robustification*, which we discuss in what follows. We use the time-series search and online bidding problems as illustrations, even though our approach may carry through in other online problems, at the expense of more complex calculations.

6.1 Resource augmentation

In this setting, we compare an *oblivious* online algorithm A with l advice bits and no information on the error bound, to an online algorithm B that has k *ideal* (i.e. error-free) advice bits. Specifically, we are interested in finding the smallest $l \geq k$ (as a function of k) for which algorithm A is at least as good as algorithm B, regardless of the advice error of A.

The following theorem shows that $O(1)$-factor resource augmentation suffices to obtain an oblivious algorithm that is at least as efficient as any algorithm that operates in the ideal setting of error-free advice, and even if a fraction $1/3 - c$ of the advice bits may be erroneous, for any constant c.

▶ **Theorem 12.** *Consider the time-series and the online bidding problems. For all sufficiently large k, and any $c \in (0, 1/3)$, there is an oblivious online algorithm A with advice of size l, whose competitive ratio is at least as good as that of* any *online algorithm B with k bits of perfect (i.e. error-free) advice, where $l = \frac{1}{(\frac{2}{3}+c)(1-\mathcal{H}(\frac{\frac{1}{3}-c}{\frac{2}{3}+c}))}k + 1$, for any error $\eta \leq (1/3 - c)l$ in the advice of A.*

6.2 Robustification

In this setting, we augment the imperfect advice framework by requiring not only that the algorithm minimizes the competitive ratio assuming that the advice error is at most the tolerance H, but also that its competitive ratio does not exceed a *robustness* requirement r, for some specified r, if the error exceeds H (and in particular, if the advice is adversarially generated). We call such online algorithms *r-robust.* Thus, this model can be seen as an extension of both the imperfect advice and the untrusted advice model of [5].

For the time-series problem, we obtain the following result, which generalizes Theorem 5. In particular, note that Theorem 5 is a special case of Theorem 13 for $\rho = 1$.

▶ **Theorem 13.** *Consider the online time series search problem, with imperfect advice of size k, tolerance $H \leq k/2$, and robustness $r = (M/m)^{\rho}$, where $\rho \in (1/2, 1]$. There is an r-robust algorithm that uses k comparison queries, and has competitive ratio at most $(M/m)^{\frac{2\rho-1}{U+1}}$, where $U = \lfloor 2^{k-H} / \left(\!\binom{k-H}{H}\!\right) \rfloor$, for any $H \leq k/2$. Moreover, no (deterministic) algorithm based on k subset queries has competitive ratio better than $(M/m)^{\frac{2\rho-1}{L+1}}$, where $L = \lceil 2^{k} / \left(\!\binom{k-H}{H}\!\right) \rceil$.*

The analysis of r-robust algorithms for online bidding is more challenging, in particular in what concerns the impossibility results. We give an overview of the approach. For the upper bound, we can follow an analysis along the lines of Theorem 7, however, each bidding sequence in the collection $\mathcal{X}_{b,2^k}$ must be individually r-robust. This is easy to enforce, and it requires that b much be such that $b^2/(b-1) \leq r$. The lower bound is more subtle: the proof follows the lines of Theorem 9, but uses the fact that if all the l sequences in $X_0, \ldots, X_{l-1}$ must be r-robust, then $\alpha_{\bar{X}}^2/(\alpha_{\bar{X}} - 1) \leq r$. We obtain the following:

▶ **Theorem 14.** *For every $r \geq 4$ there is an r-robust bidding strategy with k-bit imperfect advice that has competitive ratio at most*

$$\min_{b>1} \frac{b^{2^k+U+1}}{b^{2^k}-1}, \quad \textit{subject to } b^{2^{k+1}}/(b^{2^k}-1) \leq r, \quad \textit{and where } U = \lceil 2^H \left(\!\!\binom{k-H}{H}\!\!\right) \rceil.$$

Furthermore, every r-robust bidding strategy has competitive ratio at least

$$\min_{\alpha>1} \frac{\alpha^{2^k+L+1}}{\alpha^{2^k}-1} \quad \textit{subject to } \alpha^{2k}/(\alpha^k-1) \leq r, \quad \textit{and where } L = \lfloor \left(\!\!\binom{k}{H}\!\!\right) \rfloor.$$

References

1 Iftikhar Ahmad, Marcus Pirron, and Günter Schmidt. Analysis of threat based algorithm using different performance measures. *RAIRO: Operations Research*, 55:2393, 2021.

2 Susanne Albers, Arindam Khan, and Leon Ladewig. Improved online algorithms for knapsack and gap in the random order model. *Algorithmica*, 83(6):1750–1785, 2021.

3 Keerti Anand, Rong Ge, Amit Kumar, and Debmalya Panigrahi. A regression approach to learning-augmented online algorithms. *Advances in Neural Information Processing Systems*, 34:30504–30517, 2021.

4 Spyros Angelopoulos. Online search with a hint. In *Proceedings of the 12th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 51:1–51:16, 2021.

5 Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc P. Renault. Online computation with untrusted advice. In *Proceedings of the 11th International Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 52:1–52:15, 2020.

6 Spyros Angelopoulos and Shahin Kamali. Contract scheduling with predictions. *Journal of Artificial Intelligence Research*, 77:395–426, 2023.

7 Spyros Angelopoulos, Shahin Kamali, and Dehou Zhang. Online search with best-price and query-based predictions. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 9652–9660, 2022.

8 Hans-Joachim Böckenhauer, Dennis Komm, Rastislav Královic, and Richard Královic. On the advice complexity of the k-server problem. *J. Comput. Syst. Sci.*, 86:159–170, 2017.

9 Hans-Joachim Böckenhauer, Dennis Komm, Rastislav Královic, Richard Královic, and Tobias Mömke. Online algorithms with advice: The tape model. *Inf. Comput.*, 254:59–83, 2017.

10 Joan Boyar, Lene M. Favrholdt, Christian Kudahl, Kim S. Larsen, and Jesper W. Mikkelsen. Online algorithms with advice: A survey. *ACM Comput. Surv.*, 50(2):19:1–19:34, 2017.

11 Joan Boyar, Kim S. Larsen, and Abyayananda Maiti. A comparison of performance measures via online search. *Theoretical Computer Science*, 532:2–13, 2014.

12 Niv Buchbinder and Joseph Naor. Improved bounds for online routing and packing via a primal-dual approach. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 293–304. IEEE, 2006.

13 Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing. *Math. Oper. Res.*, 34(2):270–286, 2009.

14 Ying Cao, Bo Sun, and Danny Tsang. Optimal online algorithms for one-way trading and online knapsack problems: A unified competitive analysis. In *Proceedings of the 59th IEEE Conference on Decision and Control (CDC)*, pages 1064–1069, 2020.

15 Marek Chrobak and Claire Kenyon-Mathieu. SIGACT news online algorithms column 10: Competitiveness via doubling. *SIGACT News*, 37(4):115–126, 2006.

16 Jhoirene Clemente, Juraj Hromkovič, Dennis Komm, and Christian Kudahl. Advice complexity of the online search problem. In *Proceedings of the 27th International Workshop on Combinatorial Algorithms (IWOCA)*, pages 203–212, 2016.

17 Peter Damaschke, Phuong Hoai Ha, and Philippas Tsigas. Online search with time-varying price bounds. *Algorithmica*, 55(4):619–642, 2009.

18 Stefan Dobrev, Rastislav Královič, and Dana Pardubská. Measuring the problem-relevant information in input. *RAIRO Theor. Informatics Appl.*, 43(3):585–613, 2009.

19 Ran El-Yaniv, Amos Fiat, Richard M Karp, and Gordon Turpin. Optimal search and one-way trading online algorithms. *Algorithmica*, 30(1):101–139, 2001.

20 Yuval Emek, Pierre Fraigniaud, Amos Korman, and Adi Rosén. Online computation with advice. *Theoretical Computer Science*, 412(24):2642–2656, 2011.

21 Shmuel Gal. A general search game. *Israel Journal of Mathematics*, 12:32–45, 1972.

22 Sungjin Im, Ravi Kumar, Aditya Petety, and Manish Purohit. Parsimonious learning-augmented caching. In *Proceedsings of the 39th International Conference on Machine Learning (ICML)*, pages 9588–9601. PMLR, 2022.

23 Sungjin Im, Ravi Kumar, Mahshid Montazer Qaem, and Manish Purohit. Online knapsack with frequency predictions. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 2733–2743, 2021.

24 Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. Primal beats dual on online packing LPs in the random-order model. *SIAM J. Comput.*, 47(5):1939–1964, 2018.

25 Dennis Komm. *An Introduction to Online Computation - Determinism, Randomization, Advice*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2016.

26 Russell Lee, Jessica Maghakian, Mohammad H. Hajiesmaili, Jian Li, Ramesh K. Sitaraman, and Zhenhua Liu. Online peak-aware energy scheduling with untrusted advice. In *Proceedings of the 12th ACM International Conference on Future Energy Systems (eEnergy)*, pages 107–123. ACM, 2021.

27 Bin Li and Steven CH Hoi. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):1–36, 2014.

28 Tongxin Li, Ruixiao Yang, Guannan Qu, Guanya Shi, Chenkai Yu, Adam Wierman, and Steven H. Low. Robustness and consistency in linear quadratic control with untrusted predictions. *Proc. ACM Meas. Anal. Comput. Syst.*, 6(1):18:1–18:35, 2022.

29 Alexander Lindermayr and Nicole Megow. Repository of works on algorithms with predictions. `https://algorithms-with-predictions.github.io/about`, 2023. Accessed: 2023-04-01.

30 Julian Lorenz, Konstantinos Panagiotou, and Angelika Steger. Optimal algorithms for k-search with application in option pricing. *Algorithmica*, 55(2):311–328, 2009.

31 Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *J. ACM*, 68(4):24:1–24:25, 2021.

32 Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error correcting codes*, volume 16. Elsevier, 1977.

33 Arya Mazumdar and Barna Saha. Clustering with noisy queries. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, pages 5788–5799. NIPS, 2017.

34 Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. In *Beyond the Worst-Case Analysis of Algorithms*, pages 646–662. Cambridge University Press, 2020.

35 Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*, pages 9661–9670, 2018.

36 Ronald L. Rivest, Albert R. Meyer, Daniel J. Kleitman, Karl Winklmann, and Joel Spencer. Coping with errors in binary search procedures. *J. Comput. Syst. Sci.*, 20(3):396–404, 1980.

37 Daniel Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.

38 Bo Sun, Ali Zeynali, Tongxin Li, Mohammad Hajiesmaili, Adam Wierman, and Danny HK Tsang. Competitive algorithms for the online multiple knapsack problem with application to electric vehicle charging. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(3):1–32, 2020.

39 Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

40 Yinfeng Xu, Wenming Zhang, and Feifeng Zheng. Optimal algorithms for the online time series search problem. *Theoretical Computer Science*, 412(3):192–197, 2011.

41 Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. Budget constrained bidding in keyword auctions and online knapsack problems. In *Proceedings of the International Workshop on Internet and Network Economics (WINE)*, pages 566–576. Springer, 2008.