Randomization Can Be as Helpful as a Glimpse of the Future in Online Computation*

Jesper W. Mikkelsen†

University of Southern Denmark, Odense, Denmark
jesperwm@imada.sdu.dk

Abstract

We provide simple but surprisingly useful direct product theorems for proving lower bounds on online algorithms with a limited amount of advice about the future. Intuitively, our direct product theorems say that if b bits of advice are needed to ensure a cost of at most t for some problem, then $r \cdot b$ bits of advice are needed to ensure a total cost of at most $r \cdot t$ when solving r independent instances of the problem. Using our direct product theorems, we are able to translate decades of research on randomized online algorithms to the advice complexity model. Doing so improves significantly on the previous best advice complexity lower bounds for many online problems, or provides the first known lower bounds. For example, we show that

- A paging algorithm needs $\Omega(n)$ bits of advice to achieve a competitive ratio better than $H_k = \Omega(\log k)$, where k is the cache size. Previously, it was only known that $\Omega(n)$ bits of advice were necessary to achieve a constant competitive ratio smaller than $5/4$.
- Every $O(n^{1-\varepsilon})$-competitive vertex coloring algorithm must use $\Omega(n \log n)$ bits of advice. Previously, it was only known that $\Omega(n \log n)$ bits of advice were necessary to be optimal.

For certain online problems, including the MTS, k-server, metric matching, paging, list update, and dynamic binary search tree problem, we prove that randomization and sublinear advice are equally powerful (if the underlying metric space or node set is finite). This means that several long-standing open questions regarding randomized online algorithms can be equivalently stated as questions regarding online algorithms with sublinear advice. For example, we show that there exists a deterministic $O(\log k)$-competitive k-server algorithm with sublinear advice if and only if there exists a randomized $O(\log k)$-competitive k-server algorithm without advice.

Technically, our main direct product theorem is obtained by extending an information theoretical lower bound technique due to Emek, Fraigniaud, Korman, and Rosén [ICALP'09].

1998 ACM Subject Classification F.1.2 Models of Computation (online computation)

Keywords and phrases online algorithms, advice complexity, information theory, randomization

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.39

1 Introduction

The model of online computation deals with optimization problems where the input arrives sequentially over time. Usually, it is assumed that an online algorithm has no knowledge of future parts of the input. While this is a natural assumption, it leaves open the possibility that a tiny amount of information about the future (which might be available in practical applications) could dramatically improve the performance guarantee of an online algorithm.

---

* Most proofs have been omitted due to space restrictions. A full version of the paper containing all proofs and technical details is available at http://arxiv.org/abs/1511.05886.

† This work was partially supported by the Villum Foundation and the Danish Council for Independent Research, Natural Sciences.

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 39; pp. 39:1–39:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Recently, the notion of advice complexity [17, 29, 34, 28] was introduced in an attempt to provide a quantitative and problem-independent framework for studying *semi-online* algorithms with limited (instead of non-existing) knowledge of the future. In this framework, the limited knowledge is modeled as a number of advice bits provided to the algorithm by an oracle (see Definition 7). The goal is to determine how much advice (measured in the length, n, of the input) is needed to achieve a certain competitive ratio. In particular, one of the most important questions is how much advice is needed to break the lower bounds for (randomized) online algorithms without advice. It has been shown that for e.g. bin packing and makespan minimization on identical machines, $O(1)$ bits of advice suffice to achieve a better competitive ratio than what is possible using only randomization [5, 3]. On the other hand, for a problem such as edge coloring, it is known that $\Omega(n)$ bits of advice are needed to achieve a competitive ratio better than that of the best deterministic online algorithm without advice [41]. However, for many online problems, determining the power of a small amount of advice has remained an open problem.

With a few notable exceptions (e.g. [15, 29, 16, 20]), most of the previous research on advice complexity has been problem specific. In this paper, we take a more complexity-theoretic approach and focus on developing techniques that are applicable to many different problems. Our main conceptual contribution is a better understanding of the connection between advice and randomization. Before explaining our results in details, we briefly review the most relevant previous work.

Standard derandomization techniques imply that a randomized algorithm can be converted (maintaining its competitiveness) into a deterministic algorithm with advice complexity $O(\log\log I(n) + \log n)$, where $I(n)$ is the number of inputs of length n [16]. Clearly, there are problems where even a single bit of advice is much more powerful than randomization, and so we cannot in general hope to convert an algorithm with advice into a randomized algorithm. However, using machine learning techniques, Blum and Burch have shown that a metrical task system algorithm with advice complexity $O(1)$ can be converted into a randomized algorithm without advice [12]. Problems such as paging, k-server, and list update can be modeled as metrical task systems (see e.g. [18]).

2 Overview of results and techniques

We will give a high-level description of the results and techniques introduced in this paper. For simplicity, we restrict ourselves to the case of minimization problems[1].

Direct product theorems. Central to our work is the concept of an *r-round input distribution*. Informally, this is a distribution over inputs that are made up of r rounds such that the requests revealed in each round are selected independently of all previous rounds. Furthermore, there must be a fixed upper bound on the length of each round (see Definition 8).

As the main technical contribution of the paper, we prove direct product theorems for r-round input distributions. Intuitively, a direct product theorem says that if b bits of advice are needed to ensure a cost of at most t for each individual round, then rb bits of advice are needed to ensure a cost of at most rt for the entire input. This gives rise to a useful technique for proving advice complexity lower bounds. In particular, it follows that a linear number of advice bits are needed to get a (non-trivial) improvement over algorithms without any advice at all.

[1] All of our results (and their proofs) are easily adapted to maximization problems. See [42] for details.

We provide two different theorems formalizing the above intuitive statement in different ways: Our main direct product theorem (Theorem 10) is based on an information theoretical argument similar to that of [29]. In the full paper [42], we also provide an alternative direct product theorem based on martingale theory. In this extended abstract, we will only consider the information theoretical version.

Repeatable online problems. We combine our direct product theorems with the following very simple idea: Suppose that we have a lower bound on the competitive ratio of randomized online algorithms without advice. Often, such a lower bound is proved by constructing a hard input distribution and then appealing to Yao's principle. For some online problems, it is always possible to combine (in a meaningful way) a set of input sequences $\{\sigma_1, \ldots, \sigma_r\}$ into one long input sequence $\sigma = g(\sigma_1, \ldots, \sigma_r)$ such that serving σ essentially amounts to serving the r smaller inputs individually and adding the costs incurred for serving each of them. We say that such problems are Σ-*repeatable* (see Definition 12). For a Σ-repeatable online problem, an adversary can draw r input sequences independently at random according to some hard input distribution. This gives rise to an r-round input distribution. By our direct product theorem, an online algorithm needs linear advice (in the length of the input) to do better against this r-round input distribution than an online algorithm without advice. Thus, for Σ-repeatable online problems, we get that it is possible to translate lower bounds for randomized algorithms without advice into lower bounds for algorithms with sublinear advice. More precisely, we obtain the following theorem.

▶ **Theorem 1.** *Let* P *be a* Σ*-repeatable online minimization problem and let* c *be a constant. Suppose that for every* $\varepsilon > 0$ *and every* α*, there exists an input distribution* $p_{\alpha,\varepsilon} : I \to [0, 1]$ *with finite support such that* $\mathbb{E}_{p_{\alpha,\varepsilon}}[D(\sigma)] \geq (c - \varepsilon)\, \mathbb{E}_{p_{\alpha,\varepsilon}}[OPT(\sigma)] + \alpha$ *for every deterministic algorithm* D *without advice. Then, every randomized algorithm reading at most* $o(n)$ *bits of advice on inputs of length* n *has a competitive ratio of at least* c*.*

Much research has been devoted to obtaining lower bounds for randomized algorithms without advice. Theorem 1 makes it possible to translate many of these lower bounds into advice complexity lower bounds, often resulting in a significant improvement over the previous best lower bounds (see Table 1).

For a Σ-repeatable problem, the total cost has to be the sum of costs incurred in each individual round. It is also possible to consider another kind of repeatable problems, where the total cost is the *maximum cost* incurred in a single round. We call such problems $\vee$-repeatable. Many online coloring problems are $\vee$-repeatable. For $\vee$-repeatable problems, we show in the full version of the paper [42] that under certain conditions, a constant lower bound on the competitive ratio of *deterministic* algorithms without advice carries over to randomized algorithms, even if the randomized algorithms have advice complexity $o(n)$. The proof of this result is straightforward and does not rely on our direct product theorems for Σ-repeatable problems. However, the result improves or simplifies a number of previously known advice complexity lower bounds for $\vee$-repeatable problems.

In Table 1, we have listed most of the repeatable online problems for which lower bounds on algorithms with sublinear advice existed prior to our work, and compared those previous lower bounds with the lower bounds that we obtain in this paper. We have also included two examples of repeatable online problems for which Theorem 1 provides the first known advice complexity lower bounds.

It is evident from Table 1 that there are many repeatable online problems. On the other hand, let us mention that e.g. bin packing and makespan minimization are examples of problems which are provably not repeatable.

Table 1 New and previously best lower bounds on the competitive ratio for algorithms reading $o(n)$ bits of advice on inputs of length n. For the Σ-repeatable problems, the lower bounds in this table are obtained by combining Theorem 1 with known lower bounds for randomized online algorithms without advice. For the $\vee$-repeatable problems, the lower bounds are obtained by combining our general result on $\vee$-repeatable problems with known lower bounds for *deterministic* algorithms without advice. In both cases, references to these previously known lower bounds for online algorithms without advice are provided in the second column of the table. We refer to the full paper [42] for a more detailed explanation of the entries in Table 1, and for a comparison with the current upper bounds.
Bipartite matching and Max-SAT are maximization problems, and hence the lower bound is obtained via the maximization version of Theorem 1. For paging and reordering buffer management, k denotes the cache/buffer size. For metrical task systems, N is the number of states. For the metrical task system problem and the k-server problem, the bounds are for a worst-case metric. It is also possible to use Theorem 1 together with known lower bounds for specific metric spaces. The lower bound for unit clustering is for the one-dimensional case.

	Lower bound for algorithms with advice complexity $o(n)$			
Σ-repeatable problem	**This work**		Previous best	
Paging	$\Omega(\log k)$	[19]	$5/4$	[17]
k-Server	$\Omega(\log k)$	[19]	$3/2$	[46]
2-Server	$1 + e^{-1/2}$	[25]	$3/2$	[46]
List Update	$3/2$	[47]	$15/14$	[22]
Metrical Task Systems	$\Omega(\log N)$	[19]	$\Omega(\log N)$	[29]
Bipartite Matching	$e/(e-1)$	[37]	$1+\varepsilon$	[43]
Reordering Buffer Management	$\Omega(\log\log k)$	[1]	$1+\varepsilon$	[2]
2-Sleep States Management	$e/(e-1)$	[36]	$7/6$	[13]
Unit clustering	$3/2$	[30]	–	
Max-SAT	$3/2$	[7]	–	
$\vee$-repeatable problem	**This work**		Previous best	
Edge Coloring	2	[8]	2	[41]
$L(2,1)$-Coloring on Paths	$3/2$	[11]	$3/2$	[11]
2-Vertex-Coloring	$\omega(1)$	[9]	2	[10]

Compact online problems. Note that when translating a lower bound on randomized algorithms without advice to a lower bound on algorithms with $o(n)$ bits of advice via Theorem 1, we had to make some assumptions on the lower bound. This begs the following question: Are there online problems where every lower bound (after suitable modifications) satisfies these assumptions? In order to formally answer this question, we make the following definition.

▶ **Definition 2.** Let P be a minimization problem and let $c > 1$ be a constant such that the expected competitive ratio of every randomized P-algorithm is at least c. We say that P is *compact* if for every $\varepsilon > 0$ and every $\alpha \geq 0$, there exists an input distribution $p_{\alpha,\varepsilon}$ with finite support such that if D is a deterministic online algorithm (without advice), then $\mathbb{E}_{p_{\alpha,\varepsilon}}[\mathtt{D}(\sigma)] \geq (c-\varepsilon) \cdot \mathbb{E}_{p_{\alpha,\varepsilon}}[\mathtt{OPT}(\sigma)] + \alpha$.

If an online problem is both Σ-repeatable and compact, we get the following result: Let c be a constant and let $\varepsilon > 0$. If c is a lower bound on the competitive ratio of randomized online algorithms without advice, then $c - \varepsilon$ is also a lower bound on the competitive ratio of online algorithms with sublinear advice. Equivalently, by contraposition, the existence of a c-competitive algorithm with sublinear advice implies the existence of a $(c + \varepsilon)$-competitive randomized algorithm without advice. Combining this with existing derandomization results [16] (see also [42]) yields the following complexity theoretic equivalence between randomization and sublinear advice (previously, only the forward implication was known [16]):

▶ **Theorem 3.** *Let* P *be a compact and* Σ*-repeatable minimization problem with at most* $2^{n^{O(1)}}$ *inputs of length* n*, and let* c *be a constant independent of* n*. The following are equivalent:*
1. *For every* $\varepsilon > 0$*, there exists a randomized* $(c + \varepsilon)$*-competitive* P*-algorithm without advice.*
2. *For every* $\varepsilon > 0$*, there exists a deterministic* $(c + \varepsilon)$*-competitive* P*-algorithm with advice complexity* $o(n)$*.*

In this paper, we use a technique due to Ambühl [4] and Mömke [44] to show that the class of compact and Σ-repeatable problems contains all problems which can be modeled as a metrical task system (MTS) with a finite number of states and tasks. This means that e.g. the k-server, list update, paging, and dynamic binary search tree problem are all compact and Σ-repeatable, assuming that the underlying metric space or node set is finite. For all these problems, it is known that it is possible to achieve a constant competitive ratio with respect to the length of the input [42]. Also, the number of inputs of length n for each of these problems is at most $2^{n^{O(1)}}$ (this bound holds since when we apply Theorem 3 to e.g. the k-server problem, the metric space will be fixed and not a part of the input). Thus, for each of these problems, Theorem 3 applies. Furthermore, for all the problems just mentioned (except paging), determining the best possible competitive ratio of a randomized algorithm without advice is regarded as important open problems [18, 27]. For example, the randomized k-server conjecture says that there exists a randomized $O(\log k)$-competitive k-server algorithm [39]. Theorem 3 shows that this conjecture is equivalent to the conjecture that there exists a deterministic $O(\log k)$-competitive k-server algorithm with advice complexity $o(n)$. Currently, it is only known how to achieve a competitive ratio of $O(\log k)$ using $2n$ bits of advice [16, 45].

We also show that there are compact and Σ-repeatable problems which cannot be modeled as a MTS. One such example is the metric matching problem (on finite metric spaces) [35].

Pessimistically, Theorem 3 may be seen as a barrier result which says that (for compact and Σ-repeatable online problems) designing an algorithm with sublinear advice complexity and a better competitive ratio than the currently best randomized algorithm without advice might be very difficult. Optimistically, one could hope that this equivalence might be useful in trying to narrow the gap between upper and lower bounds on the best possible competitive ratio of randomized algorithms without advice. In all cases, Theorem 3 shows that understanding better the exact power of (sublinear) advice in online computation would be very useful.

2.1 Other applications of our direct product theorem

The previously mentioned applications of our direct product theorem treat the hard input distribution as a black-box. However, it is also possible to apply our direct product theorem to explicit input distributions. Doing so yields some interesting lower bounds which cannot be obtained via Theorem 1. In what follows, we will state and briefly discuss three such lower bounds. We refer to the full paper [42] for details and for the proofs of Theorems 4 to 6.

Repeated matrix games. Let $q \in \mathbb{N}$ and let $A \in \mathbb{R}_+^{q\times q}$ be a matrix defining a two-player zero-sum game. Let V denote the value of the game defined by A. The *repeated matrix game (RMG)* with cost matrix A is an online problem where the algorithm and adversary repeatedly plays the game defined by A. The adversary is the row-player, the algorithm is the column-player, and the matrix A specifies the cost incurred by the online algorithm in each round. This generalizes the string guessing problem [15] and the generalized matching pennies problem [29] (both of these essentially corresponds to the RMG with a $q \times q$ matrix A where $A(i, j) = 1$ if $i \neq j$ and $A(i, i) = 0$). Using our direct product theorem, we easily get that for every $\varepsilon > 0$, an online algorithm which on inputs of length n is guaranteed to incur a cost of at most $(V - \varepsilon)n$ must read $\Omega(n)$ bits of advice.

▶ **Theorem 4.** *Let* `ALG` *be an algorithm for the RMG with cost matrix A. Furthermore, let V be the value of the (one-shot) two-person zero-sum game defined by A and let $0 < \varepsilon \leq V$ be a constant. If $\mathbb{E}[\mathtt{ALG}(\sigma)] \leq (V - \varepsilon)n$ for every input σ of length n, then* `ALG` *must read at least*

$$b \geq \frac{\varepsilon^2}{2\ln(2) \cdot \|A\|_\infty^2} n = \Omega(n) \tag{1}$$

bits of advice.

Furthermore, we also show how a more careful application of our direct product theorem to some particular repeated matrix games yields good trade-off results for the exact amount of advice needed to ensure a cost of at most αn for $0 < \alpha < V$.

A better bin packing lower bound via repeated matrix games. We use our results on repeated matrix games to prove the following advice complexity lower bound for bin packing:

▶ **Theorem 5.** *Let $c < 4 - 2\sqrt{2}$ be a constant. A randomized c-competitive bin packing algorithm must read at least $\Omega(n)$ bits of advice.*

Previously, Angelopoulos et al. showed that a bin packing algorithm with a competitive ratio of $c < 7/6$ had to use $\Omega(n)$ bits of advice by a reduction from the binary string guessing problem [5, 23]. From our results on repeated matrix games, we obtain a lower bound for *weighted* binary string guessing. Using the same reduction as in [5], but reducing from weighted binary string guessing instead, we improve the lower bound for bin packing algorithms with sublinear advice to $4 - 2\sqrt{2}$. Thus, even though bin packing itself is not repeatable, we can obtain a better lower bound via a reduction from a repeated matrix game.

Superlinear lower bounds for graph coloring. We obtain the following superlinear lower bound for online graph coloring by applying our direct product theorem to an ingenious hard input distribution due to Halldórsson and Szegedy [33] (they show that a randomized graph coloring algorithm *without* advice must have a competitive ratio of at least $\Omega(n/\log^2 n)$):

▶ **Theorem 6.** *Let $\varepsilon > 0$ be a constant. A randomized $O(n^{1-\varepsilon})$-competitive online graph coloring algorithm must read at least $\Omega(n \log n)$ bits of advice.*

Previously, it was only known that $\Omega(n \log n)$ bits of advice were necessary to be 1-competitive [32]. Note that $O(n \log n)$ bits of advice trivially suffice to achieve optimality for graph coloring. Furthermore, it is not hard to prove that for every $c = n^{1-o(1)}$, there exists a c-competitive graph coloring algorithm reading $o(n \log n)$ bits of advice. Thus, our lower bound saying that $\Omega(n \log n)$ bits are needed to be $O(n^{1-\varepsilon})$-competitive for every constant $\varepsilon > 0$ is essentially tight.

3 Relation to machine learning

Theorem 3 is very closely related to previous work on combining online algorithms [12, 31, 6]. Let $A_1, \ldots, A_m$ be m algorithms for a MTS of finite diameter Δ. Based on a variant of the celebrated machine learning algorithm Randomized Weighted Majority (RWM) [40], Blum and Burch obtained the following result [12, 24]: For every $\varepsilon > 0$, it is possible to combine the m algorithms into a single randomized MTS-algorithm, R, such that

$$\mathbb{E}[R(\sigma)] = (1 + 2\varepsilon) \cdot \min_{1 \leq i \leq m} A_i(\sigma) + \left(\frac{7}{6} + \frac{1}{\varepsilon}\right) \Delta \ln m, \tag{2}$$

for every input σ. An algorithm with b bits of advice corresponds to an algorithm which runs $m = 2^b$ algorithms in parallel (and selects the best one at the end). Thus, equation (2) immediately implies that given a c-competitive MTS-algorithm with advice complexity $b = O(1)$, we can convert it to a randomized $(c + \varepsilon)$-competitive algorithm without advice.

Theorem 3 improves on the result of Blum and Burch in two ways. First of all, it allows us to convert algorithms with sublinear instead of only constant advice complexity. Furthermore, Theorem 3 applies to all compact and Σ-repeatable online problems, not just those which can be modeled as a MTS.

It is natural to ask if it is possible to use the technique of Blum and Burch in order to obtain a constructive proof of Theorem 3. To this end, we remark that the result of Blum and Burch relies fundamentally on the fact that the cost incurred by RWM when switching from the state of algorithm A_i to the state of algorithm A_j for $i \neq j$ is bounded by a constant. Thus, it does not seem possible to extend the result to those compact and Σ-repeatable problems which does not satisfy this requirement (such as the metric matching problem). On the other hand, in the full paper [42], we show that by combining the result of Blum and Burch with the ideas that we use to prove that the MTS problem is compact, it is possible to use a variant of RWM to algorithmically convert a c-competitive MTS-algorithm with advice complexity $o(n)$ (instead of just $O(1)$) into a randomized $(c + \varepsilon)$-competitive algorithm without advice. This yields a constructive version of Theorem 3 for problems that can be modeled as a MTS.

Finally, we note that very shortly after and independently of our work, Böckenhauer et al. considered applications of machine learning algorithms to the advice complexity model [14]. In [14], the authors present and analyze an algorithm called Shrinking Dartboard which is similar to the algorithm of Blum and Burch (both algorithms are based on RWM).

4 The computational model

We start by formally defining competitive analysis and advice complexity. Since we are very much interested in sublinear advice, we will use the *advice-on-tape model* [17, 34]. In this model, the algorithm is allowed to read an arbitrary number of advice bits from an advice tape. There is an alternative model, the *advice-with-request model* [29], where a fixed number of advice bits is provided along with each request.

▶ **Definition 7** (Advice complexity [17, 34] and competitive ratio [18]). The input to an online problem, P, is a sequence $\sigma = (s, x_1, \ldots, x_n)$. We say that s is the *initial state* and $x_1, \ldots, x_n$ are the *requests*. A *deterministic online algorithm with advice*, `ALG`, computes the output $\gamma = (y_1, \ldots, y_n)$, under the constraint that y_i is computed from $\varphi, s, x_1, \ldots, x_i$, where φ is the content of the advice tape. The *advice complexity*, $b(n)$, of `ALG` is the largest number of bits of φ read by `ALG` over all possible inputs of length at most n.

For an input σ, $\texttt{ALG}(\sigma)$ ($\texttt{OPT}(\sigma)$) denotes the non-negative real *cost* of the output computed by $\texttt{ALG}$ ($\texttt{OPT}$) when serving σ. We say that $\texttt{ALG}$ is *c-competitive* if there exists a constant α such that $\texttt{ALG}(\sigma) \leq c \cdot \texttt{OPT}(\sigma) + \alpha$ for all inputs σ.

A *randomized online algorithm* $\texttt{R}$ *with advice complexity* $b(n)$ is a probability distribution over deterministic online algorithms with advice complexity at most $b(n)$. We say that $\texttt{R}$ is c-competitive if there exists a constant α such that $\mathbb{E}[\texttt{R}(\sigma)] \leq c \cdot \texttt{OPT}(\sigma) + \alpha$ for all inputs σ. △

5 An information theoretical direct product theorem

In this section, we formally state (and sketch how to prove) our direct product theorem upon which all of our results rely. Given P-inputs $\sigma_1, \ldots, \sigma_r$, we define $\sigma = \sigma_1 \ldots \sigma_r$ to be the P-input obtained by concatenating the requests of the r inputs and using the same initial state as σ_1. For example, if $\sigma_1 = (s, x_1, \ldots, x_n)$ and $\sigma_2 = (s', x'_1, \ldots, x'_{n'})$, then $\sigma_1\sigma_2 = (s, x_1, \ldots, x_n, x'_1, \ldots, x'_{n'})$.

▶ **Definition 8** (r-round input distribution)**.** Let P be a minimization problem and let $r \in \mathbb{N}$. For each $1 \leq i \leq r$, let I_i be a finite set of P-inputs such that the following holds: If $\sigma_1, \ldots, \sigma_r$ are such that $\sigma_i \in I_i$ for $1 \leq i \leq r$, then $\sigma = \sigma_1\sigma_2 \ldots \sigma_r$ is a valid P-input. Furthermore, let $I^r = I_1 \times \cdots \times I_r = \{\sigma_1 \ldots \sigma_i \ldots \sigma_r \mid \sigma_i \in I_i \text{ for } 1 \leq i \leq r\}$.

For each $1 \leq i \leq r$, let cost_i be a function which maps an output γ computed for an input $\sigma \in I^r$ to a non-negative real number $\text{cost}_i(\gamma, \sigma)$. We say that cost_i is the ith round *cost function*.

Let $p_i : I_i \to [0, 1]$ be a probability distribution over I_i and let $p^r : I^r \to [0, 1]$ be the (product) probability distribution which maps $\sigma_1\sigma_2 \ldots \sigma_r \in I^r$ into $p_1(\sigma_1)p_2(\sigma_2) \cdots p_r(\sigma_r)$. We say that p^r (together with the associated cost functions cost_i) is an *r-round input distribution*. For $1 \leq i \leq r$, we say that p_i is the ith round input distribution of p^r. △

▶ **Definition 9.** Let P be a minimization problem and let $r \in \mathbb{N}$. Let $p^r : I^r \to [0, 1]$ be an r-round input distribution with associated cost functions cost_i and with ith round input distributions p_i. Let $\texttt{ALG}$ be a deterministic P-algorithm with advice. We define the following random variables: Let X be the entire input and Y the output computed by $\texttt{ALG}$. For $1 \leq i \leq r$, let X_i be the requests revealed in round i, and let $\text{cost}_i(\texttt{ALG}) = \text{cost}_i(Y, X)$ be the ith round cost function applied to the output computed by $\texttt{ALG}$. Also, let $\text{cost}(\texttt{ALG}) = \sum_{i=1}^{r} \text{cost}_i(\texttt{ALG})$, let B be the advice bits read by $\texttt{ALG}$, and let $W_i = (X_1, \ldots, X_{i-1}, B)$. Random variables are always denoted by capital letters and their support by the calligraphic version of that letter[2].

Finally, for every $1 \leq i \leq r$ and every $w \in \mathcal{W}_i$, we define the conditional ith round input distribution $p_{i|w} : I_i \to [0, 1]$ as follows:

$$p_{i|w}(x) = p_i(x|W_i = w) = \frac{\Pr(X_i = x, W_i = w)}{\Pr(W_i = w)}.$$

△

We prove our information theoretical direct product theorem by extending an entropy based lower bound technique due to Emek et al. [29]. The statement and proof of the theorem relies on basic notions and results from information theory (see [42] or [26]). In particular, given two distributions $\mu, \nu : \Omega \to [0, 1]$ such that $\text{supp}(\mu) \subseteq \text{supp}(\nu)$, the *Kullback-Leibler*

[2] For example, the support of W_i is denoted $\mathcal{W}_i$, where $\mathcal{W}_i = \{w \colon \Pr[W_i = w] > 0\}$.

divergence $D_{KL}(\mu\|\nu)$ between μ and ν is defined as $D_{KL}(\mu\|\nu) = \sum_{\omega\in\Omega} \mu(\omega)\log(\mu(\omega)/\nu(\omega))$. Also, we heavily use the notation of Definitions 8 and 9.

▶ **Theorem 10.** *Let* P *be a minimization problem and let* p^r *be an* r*-round input distribution with associated cost-functions* cost_i*. Furthermore, let* ALG *be a deterministic algorithm reading at most* b *bits of advice on every input in the support of* p^r*. Assume that there exists a convex and decreasing function* $f : [0,\infty] \to \mathbb{R}$ *such that for every* $1 \le i \le r$ *and* $w \in \mathcal{W}_i$*, the following holds:*

$$\mathbb{E}[\text{cost}_i(\texttt{ALG})|W_i = w] \ge f\left(D_{KL}(p_{i|w}\|p_i)\right). \tag{3}$$

Then, $\mathbb{E}[\text{cost}(\texttt{ALG})] \ge rf(b/r)$*.*

Before sketching the proof of Theorem 10, we informally discuss the theorem. Recall that W_i is the information available to the algorithm when round i begins (the advice read and the history of previous requests). Without any advice or knowledge of the history, the probability of $x \in \mathcal{X}_i$ being selected as the round i request sequence is $p_i(x)$. However, this probability may change given that the algorithm knows W_i (in the most extreme case, the advice could specify exactly the request sequence in round i). For any fixed $w \in \mathcal{W}_i$, the probability of x being selected in round i given that $W_i = w$ is denoted $p_{i|w}(x)$. Assumption (3) informally means that the closer $p_{i|w}$ and p_i are to each other, the better a lower bound we must have on the expected cost incurred by ALG in round i given that ALG knows w. We remark that the convexity assumption on f is automatically satisfied in most applications. The conclusion of Theorem 10 essentially says that under these assumptions, an algorithm with b bits of advice for the entire input can do no better than an algorithm with b/r bits of advice for each individual round. In particular, this will allow us to conclude that $\Omega(r)$ bits of advice are needed to get a non-trivial improvement over having no advice at all. The complete proof of Theorem 10 can be found in the full version of the paper [42].

Proof Sketch (of Theorem 10). Fix i such that $1 \le i \le r$. Using first the law of total expectation and then combining assumption (3) with Jensen's inequality, we get that

$$\mathbb{E}[\text{cost}_i(\texttt{ALG})] = \mathbb{E}_w[\mathbb{E}[\text{cost}_i(\texttt{ALG})|W_i = w]] \ge f\left(\mathbb{E}_w[D_{\text{KL}}(p_{i|w}\|p_i)]\right). \tag{4}$$

A simple calculation shows that the expected Kullback-Leibler divergence $\mathbb{E}_w[D_{\text{KL}}(p_{i|w}\|p_i)]$ equals the mutual information $I(X_i;W_i)$ between X_i and W_i. Thus,

$$\mathbb{E}[\text{cost}(\texttt{ALG})] = \sum_{i=1}^{r} \mathbb{E}[\text{cost}_i(\texttt{ALG})] \ge \sum_{i=1}^{r} f(I(X_i;W_i))$$

The mutual information $I(X;B)$ between the input X and the advice B is at most b simply because the entropy $H(B)$ of B is at most b (since ALG reads at most b bits of advice). On the other hand, using that the r rounds are independent and the chain rule of conditional entropy, we get that $I(X;B) = \sum_{i=1}^{r} I(X_i;W_i)$. Combining these two observations, we see that $I(X_1;W_1),\ldots,I(X_r,W_r)$ are non-negative real numbers which sums to at most b. Since f is convex and decreasing, Jensen's inequality then implies that $\sum_{i=1}^{r} f(I(X_i;W_i)) \ge rf(b/r)$. ◀

6 Application: Repeatable online problems

In this section we will present the main ingredients of the proof of Theorem 1. Before we begin, we need to formally define the repeated version, P^*_Σ, of an online problem P, and we need to define precisely what it means to be Σ-repeatable. To this end, given P-inputs $\sigma_1, \ldots, \sigma_r$ with the same initial state s, we define $(\sigma_1; \ldots; \sigma_r)$ to be a sequence with the requests of the r inputs concatenated and such that the initial state s arrives together with the first request of each σ_i. For example, if $\sigma_1 = (s, x_1, \ldots, x_n)$ and $\sigma_2 = (s, x'_1, \ldots, x'_{n'})$, then $(\sigma_1; \sigma_2) = (s, \{s, x_1\}, x_2, \ldots, x_n, \{s, x'_1\}, x'_2, \ldots, x'_{n'})$.

▶ **Definition 11.** Let P be an online problem, let S be the set of initial states for P, and let I (I_s) be the set of all possible request sequences for P (with initial state s). Define P^*_Σ to be the online problem with inputs $I^* = \{\sigma^* = (\sigma_1; \sigma_2; \ldots; \sigma_r) \mid r \geq 1, s \in S, \sigma_i \in I_s\}$. An algorithm for P^*_Σ must produce an output $\gamma^* = (\gamma_1, \ldots, \gamma_r)$ where $\gamma_i = (y_1, \ldots, y_{n_i})$ is a valid sequence of answers for the P-input $\sigma_i = (s, x_1, \ldots, x_{n_i}) \in I_s$. The score of the output γ^* is $\mathrm{score}(\gamma^*, \sigma^*) = \sum_{i=1}^{r} \mathrm{score}_P(\gamma_i, \sigma_i)$, where $\mathrm{score}_P(\gamma_i, \sigma_i)$ is the score of the P-output γ_i with respect to the P-input σ_i. The optimal offline algorithm for P^*_Σ is denoted $\mathtt{OPT}^*_\Sigma$. △

$\mathrm{P}^*_\vee$ is defined similarly, except that $\mathrm{score}(\sigma^*, \gamma^*) = \max\{\mathrm{score}_P(\sigma_1, \gamma_1), \ldots, \mathrm{score}_P(\sigma_r, \gamma_r)\}$.

In order to better understand the definition of P^*_Σ, it is useful to imagine that after serving the last request of round $i-1$ but before serving the first request of round i, the current state is changed to the initial state s (note that the algorithm knows when this happens since in $(\sigma_1; \ldots; \sigma_r)$, the first request of each σ_i is special). For instance, if P is the k-server problem, then an initial state s is a placement of the k servers in the metric space. Thus, in P^*_Σ, after serving the last request of round $i-1$ and before serving the first request of round i, the k servers automatically return to their initial position specified by s. Note that when P is the k-server problem, we can concatenate P-inputs $\sigma_1, \sigma_2, \ldots, \sigma_r$ into one long P-input $\sigma = \sigma_1\sigma_2 \ldots \sigma_r$. The only difference between serving the P-input σ and serving the P^*_Σ-input $\sigma^* = (\sigma_1; \ldots; \sigma_r)$ is that for the P-input σ, the k servers are not returned to the initial state s when a round ends. However, if the underlying metric space has finite diameter, this difference in the placement of servers when a new round begins cannot invalidate a lower bound. In fact, it turns out that for many online problems, there is a natural reduction from P^*_Σ to P that essentially preserves all lower bounds. This is formalized in Definition 12.

▶ **Definition 12.** Let P be an online minimization problem such that for every fixed P-input, there is only a finite number of valid outputs. Furthermore, let $k_1, k_2, k_3 \geq 0$. We say that P is Σ-*repeatable with parameters* (k_1, k_2, k_3) if there exists a mapping $g : I^* \to I$ with the following properties:

Σ1. $|g(\sigma^*)| \leq |\sigma^*| + k_1 r$ for every $\sigma^* \in I^*$ consisting of r rounds.

Σ2. For every deterministic P-algorithm $\mathtt{ALG}$, there exists a deterministic P^*_Σ-algorithm $\mathtt{ALG}^*$ such that $\mathtt{ALG}^*(\sigma^*) \leq \mathtt{ALG}(g(\sigma^*)) + k_2 r$ for every $\sigma^* \in I^*$ consisting of r rounds.

Σ3. $\mathtt{OPT}^*_\Sigma(\sigma^*) \geq \mathtt{OPT}(g(\sigma^*)) - k_3 r$ for every $\sigma^* \in I^*$ consisting of r rounds. △

The definition of ∨-repeatable is identical to that of Σ-repeatable, except that P^*_Σ and $\mathtt{OPT}^*_\Sigma$ are replaced by $\mathrm{P}^*_\vee$ and $\mathtt{OPT}^*_\vee$. The k-server problem on a metric space of diameter Δ is Σ-repeatable with parameters $(0, k\Delta, k\Delta)$.

We will now sketch the proof of Theorem 1. The complete proof is in the full paper [42].

Proof Sketch (of Theorem 1). We prove the desired lower bound for P^*_Σ. Let ε and α be arbitrary. Without loss of generality, we may assume that all inputs in the support of $p_{\alpha,\varepsilon}$ have the same initial state. Draw r inputs independently from $p_{\alpha,\varepsilon}$. This gives rise to an r-round input distribution $p^r_{\alpha,\varepsilon}$ for P^*_Σ. Note that the ith round input distribution, p_i, of $p^r_{\alpha,\varepsilon}$ is simply $p_i = p_{\alpha,\varepsilon}$, and that a round of $p^r_{\alpha,\varepsilon}$ corresponds to a round of P^*_Σ.

Let $\texttt{ALG}^*$ be a deterministic P_Σ^*-algorithm reading at most b bits of advice on inputs in $\mathrm{supp}(p_{\alpha,\varepsilon}^r)$. Fix a round $1 \le i \le r$ and let $w \in \mathcal{W}_i$. We need to give a lower bound on the expected cost $\mathbb{E}[\mathrm{cost}_i(\texttt{ALG}^*)|W_i = w]$ in terms of the Kullback-Leibler divergence between $p_{i|w}$ and p_i. To this end, we use *Pinsker's inequality*. Pinsker's inequality ([26, Lemma 11.6.1]) says this if the Kullback-Leibler divergence between $p_{i|w}$ and p_i is small, then $p_{i|w}$ and p_i must be close in L_1-norm: $\|p_{i|w} - p_i\|_1 \le \sqrt{\ln 4 \cdot D_{\mathrm{KL}}(p_{i|w} \| p_i)}$. Thus, it suffices to bound $\mathbb{E}[\mathrm{cost}_i(\texttt{ALG}^*)|W_i = w]$ in terms of the L_1-distance between $p_{i|w}$ and p_i.

To this end, we construct a P-algorithm, $\texttt{ALG}_w$, without advice by hard-wiring w (i.e., the advice and the requests in rounds 1 to $i-1$) into the P_Σ^*-algorithm $\texttt{ALG}^*$. That is, for an input sequence $\sigma \in \mathrm{supp}(p_{i|w})$, the new algorithm $\texttt{ALG}_w$ simulates the computation of $\texttt{ALG}^*$ on σ when $W_i = w$ and $\texttt{ALG}^*$ is given the requests σ in round i. Note that this is possible since w is fixed, and hence the output of $\texttt{ALG}^*$ in round i given σ as input in this round is completely determined. For all other input sequences, $\texttt{ALG}_w$ behaves arbitrarily (but does compute some valid output). It follows that $\texttt{ALG}_w$ is well-defined for all input sequences in $\mathrm{supp}(p_i)$. Thus, $\texttt{ALG}_w$ defines a mapping $\sigma \mapsto \texttt{ALG}_w(\sigma)$ on $\mathrm{supp}(p_i)$ such that $\|\texttt{ALG}_w\|_\infty = \max_{\sigma \in \mathrm{supp}(p_i)} |\texttt{ALG}_w(\sigma)| \le M$ and such that if $\sigma \in \mathrm{supp}(p_{i|w}) \subseteq \mathrm{supp}(p_i)$, then $\texttt{ALG}_w(\sigma)$ is equal to the cost incurred by $\texttt{ALG}^*$ if $W_i = w$ and σ is given as input in round i. It follows that

$$\begin{aligned}\mathbb{E}[\mathrm{cost}_i(\texttt{ALG})|W_i = w] &= \mathbb{E}_{\sigma \sim p_{i|w}}[\texttt{ALG}_w(\sigma)] \ge \mathbb{E}_{\sigma \sim p_i}[\texttt{ALG}_w(\sigma)] - \|\texttt{ALG}_w\|_\infty \cdot \|p_{i|w} - p_i\|_1 \\ &\ge (c-\varepsilon)\,\mathbb{E}_{\sigma \sim p_i}[\texttt{OPT}(\sigma)] + \alpha - M \cdot \sqrt{\ln 4 \cdot D_{\mathrm{KL}}(p_{i|w} \| p_i)}.\end{aligned}$$

Let $f(d) = (c-\varepsilon)\,\mathbb{E}_{\sigma \sim p_i}[\texttt{OPT}(\sigma)] + \alpha - M \cdot \sqrt{\ln 4 \cdot d}$. Note that f is convex. Our direct product theorem (Theorem 10) therefore yields (remember that $p_i = p_{\alpha,\varepsilon}$).

$$\mathbb{E}[\mathrm{cost}(\texttt{ALG}^*)] \ge r f(b/r) \ge r\left((c-\varepsilon)\,\mathbb{E}_{\sigma \sim p_{\alpha,\varepsilon}}[\texttt{OPT}(\sigma)] + \alpha - 2M\sqrt{\frac{b}{r}}\right). \tag{5}$$

Assume that $\texttt{ALG}^*$ uses sublinear advice, i.e., $b = o(n)$. Note that $n = \Theta(r)$ since $\mathrm{supp}(p_{\alpha,\varepsilon})$ is finite. Thus, we can make $2M\sqrt{b/r}$ arbitrarily small by choosing the number of rounds r sufficiently large. Furthermore, by linearity of expectation, $r\,\mathbb{E}_{\sigma \sim p_{\alpha,\varepsilon}}[\texttt{OPT}(\sigma)] = \mathbb{E}[\mathrm{cost}(\texttt{OPT}_\Sigma^*)]$. Thus, since ε and α was arbitrary, it follows from (5) and Yao's principle that a randomized P_Σ^*-algorithm with sublinear advice must have a competitive ratio of at least c. Using that P is Σ-repeatable, it is possible to convert this lower bound into a lower bound for P. ◀

7 Application: Compact online problems

Recall (Definition 2) that we defined compact online problems to be those for which Theorem 1 could be used to obtain tight lower bounds. Theorem 3 follows trivially by combining Definition 2 with Theorem 1 (and using previously known derandomization results). In this section, we will sketch how to prove that several important Σ-repeatable online problems are compact. Interestingly, our proof will rely on the "upper bound part" of Yao's principle [48] which is (much) less frequently used than the lower bound part.

Fix a Σ-repeatable problem P such that for every n (and every fixed initial state), the number of inputs of length at most n is finite[3]. Let $c > 1$ be a constant such that the expected

[3] Since P is Σ-repeatable, this assumption automatically implies that there is only finitely many P-algorithms for inputs of length at most n.

competitive ratio of every randomized P-algorithm is at least c. What does it mean for P to *not* be compact? It means that there exists an $\varepsilon > 0$ and $\alpha \geq 0$ such that the following holds: For every $n' \in \mathbb{N}$ and for every probability distribution p over P-inputs of length at most n', there exists a deterministic algorithm D such that $\mathbb{E}_{\sigma\sim p}[\mathtt{D}(\sigma)] < (c-\varepsilon)\,\mathbb{E}_{\sigma\sim p}[\mathtt{OPT}(\sigma)] + \alpha$. Recall that, by assumption, there is only a finite number of inputs and outputs for P of length at most n'. This makes it possible to view the problem as a *finite* two-player zero-sum game between an algorithm and adversary. Thus (see the full paper [42] for a formal proof), by Yao's principle, we get that there exists a randomized P-algorithm $\mathtt{R}_{n'}$ such that $\mathbb{E}[\mathtt{R}_{n'}(\sigma)] < (c-\varepsilon)\mathtt{OPT}(\sigma)+\alpha$ for every P-input of length at most n'. Now, if we can somehow show that it is possible to use the algorithms $\mathtt{R}_1, \mathtt{R}_2, \ldots$ to obtain a single algorithm R which is better than c-competitive (on all possible inputs), then the problem at hand must, by contradiction, be compact.

From the above discussion, it is easy to see that the metric matching problem is compact (on finite metric spaces). Indeed, in this problem, there are k servers placed in a metric space. Each server can be matched to at most one request (and vice versa). But this means that for every fixed metric space and fixed set of k servers, the length of the input never exceeds k. This allow us to construct a single algorithm R which on inputs with k servers run the appropriate algorithm $\mathtt{R}_k$. The algorithm R will be $(c-\varepsilon)$-competitive. This argument extends to all Σ-repeatable problems for which an online algorithm knows a priori some upper bound on the number of requests.

Another important example is the k-server problem on a finite metric space. In this problem, we also have a metric space and a set of k servers. However, for the k-server problem, there is no bound on the number of times we may use a single server. Thus, the length of the input is unbounded, even if we fix the metric space and the set of servers. This means that we cannot just trivially run the appropriate algorithm $\mathtt{R}_{n'}$, since we do not know an upper bound n' on the length of the input. However, what we can do is the following (see the full paper and [4, 44]): Choose n' to be some sufficiently large number. If the number of requests exceeds n', we *restart* $\mathtt{R}_{n'}$. That is, we use a new instantiation of $\mathtt{R}_{n'}$ which pretends that the $(n'+1)$'th request is in fact the very first request of the input sequence. By appropriately handling some technical issues, we can make sure that the price of performing these restarts is sufficiently small compared to the cost of an optimal solution. This gives a single algorithm R which is better than c-competitive on inputs of arbitrary length. Thus, the k-server problem is compact. This technique works for all problems that can be modeled as a MTS. We refer to the full version of the paper for more details.

8 Further work and open problems

We have attempted to make the results and techniques introduced in this paper as easy as possible to apply and build on. Komm et al. have used our results on repeated matrix games to prove lower bounds for certain online hereditary graph problems with preemption [38]. Also, in Boyar et al. [21], we have applied Theorem 1 to online weighted matching.

Currently, the equivalence between advice and randomization stated in Theorem 3 is mainly being used to obtain knowledge about algorithms with advice using techniques and results for randomized algorithms. It is an interesting open problem to what extent the equivalence is also useful in the other direction.

Acknowledgment. The author thanks Joan Boyar, Magnus Find, Lene Favrholdt, and the anonymous ICALP reviewers for helpful comments on this work and its presentation.

References

1 Anna Adamaszek, Artur Czumaj, Matthias Englert, and Harald Räcke. Almost tight bounds for reordering buffer management. In *STOC*, 2011.

2 Anna Adamaszek, Marc P. Renault, Adi Rosén, and Rob van Stee. Reordering buffer management with advice. In *WAOA*, 2013.

3 Susanne Albers and Matthias Hellwig. Online makespan minimization with parallel schedules. In *SWAT*, 2014.

4 Christoph Ambühl. *On the list update problem*. PhD thesis, ETH Zürich, 2002.

5 Spyros Angelopoulos, Christoph Dürr, Shahin Kamali, Marc P. Renault, and Adi Rosén. Online bin packing with advice of small size. In *WADS*, 2015.

6 Yossi Azar, Andrei Z. Broder, and Mark S. Manasse. On-line choice of on-line algorithms. In *SODA*, 1993.

7 Yossi Azar, Iftah Gamzu, and Ran Roth. Submodular max-sat. In *ESA*, 2011.

8 Amotz Bar-Noy, Rajeev Motwani, and Joseph (Seffi) Naor. The greedy algorithm is optimal for on-line edge coloring. *Inf. Process. Lett.*, 44(5):251–253, 1992.

9 Dwight R. Bean. Effective coloration. *J. Symb. Log.*, 41(2):469–480, 1976.

10 Maria Paola Bianchi, Hans-Joachim Böckenhauer, Juraj Hromkovič, and Lucia Keller. Online coloring of bipartite graphs with and without advice. *Algorithmica*, 70(1):92–111, 2014.

11 Maria Paola Bianchi, Hans-Joachim Böckenhauer, Juraj Hromkovič, Sacha Krug, and Björn Steffen. On the advice complexity of the online $L(2,1)$-coloring problem on paths and cycles. *Theor. Comput. Sci.*, 554:22–39, 2014.

12 Avrim Blum and Carl Burch. On-line learning and the metrical task system problem. *Machine Learning*, 39(1):35–58, 2000.

13 Hans-Joachim Böckenhauer, Richard Dobson, Sacha Krug, and Kathleen Steinhöfel. On energy-efficient computations with advice. In *COCOON*, 2015.

14 Hans-Joachim Böckenhauer, Sascha Geulen, Dennis Komm, and Walter Unger. Constructing randomized online algorithms from algorithms with advice. *ETH-Zürich*, 2015.

15 Hans-Joachim Böckenhauer, Juraj Hromkovič, Dennis Komm, Sacha Krug, Jasmin Smula, and Andreas Sprock. The string guessing problem as a method to prove lower bounds on the advice complexity. *Theor. Comput. Sci.*, 554:95–108, 2014.

16 Hans-Joachim Böckenhauer, Dennis Komm, Rastislav Královič, and Richard Královič. On the advice complexity of the k-server problem. In *ICALP (1)*, 2011.

17 Hans-Joachim Böckenhauer, Dennis Komm, Rastislav Královič, Richard Královič, and Tobias Mömke. On the advice complexity of online problems. In *ISAAC*, 2009.

18 Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

19 Allan Borodin, Nathan Linial, and Michael E. Saks. An optimal on-line algorithm for metrical task system. *J. ACM*, 39(4):745–763, 1992.

20 Joan Boyar, Lene M. Favrholdt, Christian Kudahl, and Jesper W. Mikkelsen. Advice complexity for a class of online problems. In *STACS*, 2015.

21 Joan Boyar, Lene M. Favrholdt, Christian Kudahl, and Jesper W. Mikkelsen. Weighted online problems with advice. In submission, 2016.

22 Joan Boyar, Shahin Kamali, Kim S. Larsen, and Alejandro López-Ortiz. On the list update problem with advice. In *LATA*, 2014.

23 Joan Boyar, Shahin Kamali, Kim S. Larsen, and Alejandro López-Ortiz. Online bin packing with advice. In *STACS*, 2014. Full paper to appear in *Algorithmica*.

24 Carl Burch. *Machine learning in metrical task systems and other on-line problems*. PhD thesis, Carnegie Mellon University, 2000. http://cburch.com/pub/thesis.ps.gz.

25 Marek Chrobak, Lawrence L. Larmore, Carsten Lund, and Nick Reingold. A better lower bound on the competitive ratio of the randomized 2-server problem. *Inf. Process. Lett.*, 63(2):79–83, 1997.

26 Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley, 2006.

27 Erik D. Demaine, Dion Harmon, John Iacono, and Mihai Patrascu. Dynamic optimality – almost. *SIAM J. Comput.*, 37(1):240–251, 2007.

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

29 Yuval Emek, Pierre Fraigniaud, Amos Korman, and Adi Rosén. Online computation with advice. *Theor. Comput. Sci.*, 412(24):2642–2656, 2011.

30 Leah Epstein and Rob van Stee. On the online unit clustering problem. *ACM Transactions on Algorithms*, 7(1):1–18, 2010.

31 Amos Fiat, Dean P. Foster, Howard J. Karloff, Yuval Rabani, Yiftach Ravid, and Sundar Vishwanathan. Competitive algorithms for layered graph traversal. *SIAM J. Comput.*, 28(2):447–462, 1998.

32 Michal Forišek, Lucia Keller, and Monika Steinová. Advice complexity of online graph coloring. Unpublished manuscript, 2012.

33 Magnús M. Halldórsson and Mario Szegedy. Lower bounds for on-line graph coloring. *Theor. Comput. Sci.*, 130(1):163–174, 1994.

34 Juraj Hromkovič, Rastislav Královič, and Richard Královič. Information complexity of online problems. In *MFCS*, 2010.

35 Bala Kalyanasundaram and Kirk Pruhs. Online weighted matching. *J. Algorithms*, 14(3):478–488, 1993.

36 Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan S. Owicki. Competitive randomized algorithms for non-uniform problems. In *SODA*, 1990.

37 Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, 1990.

38 Dennis Komm, Rastislav Královic, Richard Královic, and Christian Kudahl. Advice complexity of the online induced subgraph problem. *CoRR*, abs/1512.05996, 2015.

39 Elias Koutsoupias. The k-server problem. *Computer Science Review*, 3(2):105–118, 2009.

40 Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

41 Jesper W. Mikkelsen. Optimal online edge coloring of planar graphs with advice. In *CIAC*, 2015.

42 Jesper W. Mikkelsen. Randomization can be as helpful as a glimpse of the future in online computation. *CoRR*, abs/1511.05886, November 2015.

43 Shuichi Miyazaki. On the advice complexity of online bipartite matching and online stable marriage. *Inf. Process. Lett.*, 114(12):714–717, 2014.

44 Tobias Mömke. A competitive ratio approximation scheme for the k-server problem in fixed finite metrics. *CoRR*, abs/1303.2963, 2013.

45 Marc P. Renault and Adi Rosén. On online algorithms with advice for the k-server problem. *Theory Comput. Syst.*, 56(1):3–21, 2015.

46 Jasmin Smula. *Information Content of Online Problems, Advice versus Determinism and Randomization*. PhD thesis, ETH Zürich, 2015.

47 Boris Teia. A lower bound for randomized list update algorithms. *Inf. Process. Lett.*, 47(1):5–9, 1993.

48 Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *FOCS*, 1977.