



HAL
open science

Online Search with a Hint

Spyros Angelopoulos

► **To cite this version:**

Spyros Angelopoulos. Online Search with a Hint. Information and Computation, In press, 295, pp.105091. 10.1016/j.ic.2023.105091 . hal-04248722v1

HAL Id: hal-04248722

<https://hal.science/hal-04248722v1>

Submitted on 18 Oct 2023 (v1), last revised 14 Dec 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Online Search With a Hint^{☆,☆☆}

Spyros Angelopoulos

*CNRS and LIP6–Sorbonne University
4 place Jussieu 75252, Paris, France*

Abstract

We introduce the study of search problems, in a setting in which the searcher has some information, or *hint* concerning the hiding target. In particular, we focus on one of the fundamental problems in search theory, namely the *linear search* problem. Here, an immobile target is hidden at some unknown position on an unbounded line, and a mobile searcher, initially positioned at some specific point of the line called the *root*, must traverse the line so as to locate the target. The objective is to minimize the worst-case ratio of the distance traversed by the searcher to the distance of the target from the root, which is known as the *competitive ratio* of the search.

We consider three settings in regards to the nature of the hint: i) the hint suggests the exact position of the target on the line; ii) the hint suggests the direction of the optimal search (i.e., to the left or the right of the root); and iii) the hint is a general k -bit string that encodes some information concerning the target. Our objective is to study the *Pareto*-efficiency of strategies in this model, with respect to the tradeoff between *consistency* and *robustness*. Namely, we seek optimal, or near-optimal tradeoffs between the searcher’s performance if the hint is correct (i.e., provided by a trusted source) and if the hint is incorrect (i.e., provided by an adversary).

We prove several results in each of these three settings. For positional hints, we show that the optimal consistency of r -robust strategies is $(b_r + 1)/(b_r - 1)$, where b_r is defined to be equal to $\frac{\rho_r + \sqrt{\rho_r^2 - 4\rho_r}}{2}$, and $\rho_r = (r - 1)/2$, for all $r \geq 9$. For directional hints, we show that for every $b \geq 1$ and $\delta \in (0, 1]$, there exists a strategy with consistency equal to $c = 1 + 2(\frac{b^2}{b^2-1} + \delta \frac{b^3}{b^2-1})$ and robustness equal to $1 + 2(\frac{b^2}{b^2-1} + \frac{1}{\delta} \frac{b^3}{b^2-1})$; furthermore, we show again that this upper bound is tight. Last, for general k -bit hints, we show upper bounds for general k -bit hints, as well as lower bounds: specifically, we show that the consistency of any 9-robust strategy must be at least 5, and that the consistency of r -robust strategies is at least $1 + 2b_r/(b_r - 1)$, in the case of a natural class of *asymptotic* strategies.

[☆]This research benefited from support of the FMJH Program PGMO and from the support to this program from Thales-EDF-Orange.

^{☆☆}A preliminary version of this work appeared in the Proceedings of the 12th Innovations in Theoretical Computer Science Conference (ITCS) [2].

Email address: spyros.angelopoulos@lip6.fr (Spyros Angelopoulos)

Keywords: Search problems, linear search, competitive analysis, consistency, robustness.

1. Introduction

Searching for a hidden target is a common task in everyday life, and an important computational problem with numerous applications. Problems involving search arise in diverse areas such as drilling for oil in multiple sites, the forest service looking for missing backpackers, search-and-rescue operations in the open seas, and navigating a robot between two points on a terrain [1]. All these problems involve a mobile *searcher* which must locate an immobile *target*, often also called *hider*, that lies in some unknown point in the *search domain*, i.e., the environment in which the search takes place. The searcher starts from some initial placement within the domain, denoted by O , which we call the *root*. There is, also, some underlying concept of quality of search, in the sense that we wish, in informal terms, for the searcher to be able to locate the target as efficiently as possible.

One of the simplest, yet fundamental search problems is searching on an infinite line that is unbounded both to the left and to the right of the root. In this problem, which goes back to Bellman [17] and Beck and Newman [10], the objective is to find a search strategy that minimizes the *competitive ratio* of search. More precisely, let S denote the *search strategy*, i.e., the sequence of moves that the searcher performs on the line. Given a target t , let $d(S, t)$ denote the total distance that the searcher has traveled up to the time it locates the target, and $d(t)$ the distance of t from O . We define the competitive ratio of S as

$$\text{cr}(S) = \sup_t \frac{d(S, t)}{d(t)}.$$

A strategy of minimum competitive ratio is called *optimal*. The problem of optimizing the competitive ratio of search on the line is known as the *linear search* problem (mostly within Mathematics and Operations Research), but is also known in Computer Science as the *cow path* problem.

It has long been known that the optimal (deterministic) competitive ratio of linear search is 9 [15], and is derived by a simple *doubling* strategy. Specifically, let the two semi-infinite branches of the line be labeled with 0, 1 respectively. Then in iteration i , with $i \in \mathbb{N}$, the searcher starts from O , traverses branch $i \bmod 2$ to distance 2^i , and returns to the root.

Linear search, and its generalization, the m -ray search problem, in which the search domain consists of m semi-infinite branches have been studied in several settings. Substantial work on linear search was done in the '70s and '80s predominantly by Beck and Beck, see e.g., [11, 16, 12, 13, 14]. Gal showed that a variant of the doubling strategy is optimal for m -ray search [29, 30]. These results were later rediscovered and extended in [9].

Other related work includes the study of randomization [52] and [35]; multi-searcher strategies [44]; searching with turn cost [23, 3]; the variant in which some probabilistic information on the target is known [32, 33]; the related problem of designing *hybrid algorithms* [34]; searching with an upper bound on the distance of the target from the root [43] and [20]; fault tolerant search [22, 39]; and performance measures beyond the competitive

ratio [36, 47, 4]. Competitive analysis has been applied beyond the linear and star search, for example in searching within a graph [38, 26, 6].

1.1. Searching with a hint

Previous work on competitive analysis of deterministic search strategies has mostly assumed that the searcher has no information about the target, whose position is adversarial to the search. In practice, however, we expect that the searcher may indeed have some information concerning the target. For instance, in a search-and-rescue mission, there may be some information on the last sighting of the missing person, or the direction the person had taken when last seen. The question then is: how can the searcher leverage such information, and to what possible extent?

If the hint comes from a source that is trustworthy, that is, if the hint is guaranteed to be correct, then the performance of search can improve dramatically. For example, in our problem, if the hint is the branch on which the target lies, then the optimal search is to explore that branch until the target is found, and the competitive ratio is 1. There is, however, an obvious downside: if the hint is incorrect, the search may be woefully inefficient since the searcher will walk eternally on the wrong branch, and the competitive ratio in this case is unbounded.

We are thus interested in analyzing the efficiency of search strategies in a setting in which the hint may be compromised. To this end, we first need to define formally the concept of the hint, as well as an appropriate performance measure for the search strategy. In general, the hint h is a binary string of size k , for a given $k \in \mathbb{N}^+$, where the i -th bit is a response to a query Q_i . For example, one can define a single query Q as “Is the target within distance at most 100 from O ?” and a one-bit hint, so that the hint answers the corresponding query. For another example, if Q = “Is the target to the left or to the right of O ?”, then a 1-bit hint informs the searcher about the direction it should pursue. From the point of view of upper bounds (positive results), we are interested in settings in which the queries and the associated hints have some natural interpretation, such as the ones given above. From the point of view of lower bounds (impossibility results), we are interested on the limitations of general k -bit hint strings which may be associated with *any* query, as we will discuss in more detail later.

Concerning the second issue, namely evaluating the performance of a search strategy S with a hint h , note first that S is a function of h . We will analyze the competitiveness of $S(h)$ in a model in which the competitive ratio is not defined by a single value, but rather by a pair $(c_{S,h}, r_{S,h})$. The value $c_{S,h}$ describes the competitive ratio of $S(h)$ assuming that h is *trusted*, and thus guaranteed to be correct. The value $r_{S,h}$ describes the competitive ratio of $S(h)$ when the hint is given by an *adversarial* source. More formally, we define

$$c_S = \inf_h \sup_t \frac{d(S(h), t, h)}{d(t)}, \quad \text{and} \quad r_S = \sup_t \sup_h \frac{d(S(h), t, h)}{d(t)}, \quad (1)$$

where $d(S(h), t, h)$ denotes the distance traversed in $S(h)$ for locating a target t with a hint h . We will call $c_{S,h}$ the *consistency* of $S(h)$, and $r_{S,h}$ the *robustness* of $S(h)$. To simplify

notation, we will often write S instead of $S(h)$ when it is clear from context that we refer to a strategy with a hint h .

For example, if the hint h is the branch on which the target lies, then the strategy that always trusts the hint is $(1, \infty)$ competitive, whereas the strategy that ignores the hint entirely is $(9, 9)$ -competitive. Our objective is then to find strategies that are provably *Pareto-optimal* or *Pareto-efficient* [28] in this model, and thus identify the strategies with the best tradeoff between robustness and consistency. Specifically, we say that a strategy with robustness r and consistency $f(r)$, for some function f , is Pareto-optimal if any other strategy with robustness at most r , has consistency at least $f(r)$, for all $r \geq 9$. Identifying the function f reveals the Pareto frontier of the bi-objective optimization problem (where the two objectives are the consistency and the robustness). When it is not obvious how to identify the function f , we rely to approximations, i.e., one aims to provide upper and lower bound on $f(r)$. In our results, we assume that the robustness of the strategy is bounded by some given $r \geq 9$, and our goal is to minimize the consistency. This is because an upper bound on the robustness provides us with significant information about the worst-case behavior of the strategy, which we can leverage both in terms of the upper and the lower bound analysis of the consistency (see e.g, Lemma 4, in which we obtain lower bounds on the search lengths of the strategy).

Our model is an adaptation, to search problems, of the *untrusted advice* framework for on-line algorithms proposed in [45] (which introduced the concepts of consistency and robustness in the context of paging), [50] (which applied the framework to other online problems) and [5] (which studied an extension of the advice complexity model to include erroneous advice); see also the survey [48]. Since then, studies of trade-offs between consistency and robustness have become prominent in the context of online optimization problems with machine-learned predictions, see e.g., [53, 54, 42, 40, 41].

1.2. Contribution

In this work we study the power of limitations of linear search with hints. Let $r \geq 9$ be a parameter that in general will denote the robustness of a search strategy, and let b_r be defined as

$$b_r = \frac{\rho_r + \sqrt{\rho_r^2 - 4\rho_r}}{2}, \quad \text{where } \rho_r = (r - 1)/2. \quad (2)$$

Here, b_r is the largest value for which a geometric strategy with base b_r is r -competitive (see Section 2 for details).

We consider the following classes of hints:

- *The hint is the position of the target.* Here, the hint describes the exact location of the target on the line: its distance from O , along with the branch (0 or 1) on which it lies. We present a strategy that is $(\frac{b_r-1}{b_r+1}, r)$ -competitive, and we prove it is Pareto-optimal.
- *The hint is the branch on which the target lies.* Here, the hint is information on whether the searcher is to the left or to the right of the root. We present a strategy that, given parameters $b > 1$ and $\delta \in (0, 1)$, has consistency $c = 1 + 2 \cdot (\frac{b^2}{b^2-1} + \delta \frac{b^3}{b^2-1})$, and robustness $r = 1 + 2 \cdot (\frac{b^2}{b^2-1} + \frac{1}{\delta} \frac{b^3}{b^2-1})$. Again, we prove that this strategy is Pareto-optimal.

- *The hint is a general k -bit string.* In the previous settings, the hint is a single bit, which answers the corresponding query. Here we address the question: how powerful can be a single-bit hint, or more generally a k -bit hint? In other words, how powerful can k binary queries be for linear search? We give several upper and lower bounds on the competitiveness of strategies in this setting. First, we look at the case of a single-bit hint. Here, we give a 9-robust strategy that has consistency at most $1 + 4\sqrt{2}$, whereas we show that no 9-robust strategy can have consistency less than 5, for *any* associated query. For general robustness r , we give upper and lower bounds that apply to some specific, but broadly used class of strategies, including *geometric* strategies (see Section 2 for a definition and Theorem 12 for the statement of the result). For general k , and for a given $r \geq 9$, we give an r -robust strategy whose consistency decreases rapidly as function of k (Proposition 13).

The above setting focuses on the trade-off between the consistency and the robustness of the search strategies. In Section 6, we expand our study so as to consider the effect of the *error* in hints. We discuss the challenges one has to face, and how to extend the strategies so as to make them more robust to errors.

In terms of techniques, for the first setting described above (in which the hint is the position of the target), the main idea is to analyze a geometric strategy with “large” base, namely b_r , for $r \geq 9$. The technical difficulty here is the lower bound; to this end, we prove a lemma that shows, intuitively, that for any r -robust strategy, the search length of the i -th iteration cannot be too big compared to the previous search lengths (Lemma 4). This technical result may be helpful in more broad settings (e.g., we also apply it in the setting in which the advice is a general k -bit string).

Concerning the second setting, in which the hint describes the branch, we rely on tools developed by Schuierer [51] for lower-bounding the performance of search strategies; more precisely on a theorem for lower-bounding the supremum of a sequence of functionals. But unlike [51], we use the theorem in a *parameterized* manner, that allows us to express the tradeoffs between the consistency and the robustness of a strategy, instead of their average.

Concerning the third, and most general setting, our upper bounds (i.e., the positive results) come from a strategy that has a natural interpretation: it determines a partition of the infinite line into 2^k *subsets*, and the hint describes the partition in which the target lies. The lower bounds (negative results) come from information-theoretic arguments, as is typical in the field of *advice complexity* of online algorithms (see, e.g., the survey [21]).

The broader objective of this work is to initiate the study of search games with some limited, but potentially untrusted information concerning the target. As we will show, the problem becomes challenging even in a simple search domain such as the infinite line. The framework should be readily applicable to other search games, and the analysis need not be confined to the competitive ratio, or to worst-case analysis. For example, search games in bounded domains are often studied assuming a probability distribution on the target, with the objective to minimize the expected search time (for several such examples see the book [1]). However, very little work has addressed the setting in which the searcher may have access to hints, such as the *High-Low* search games described in Section 5.2 of [1], in which a searcher wants to locate a hider on the unit interval by a sequence of guesses. Again,

our model is applicable, in that one would like to find the best tradeoff on the expected time to locate the target assuming a trusted or untrusted hint.

1.3. Further related work

There are several works related to exploring a known or unknown environment in the standard advice complexity model. Here, the explorer is enhanced with an additional information called *advice*, which is guaranteed to be error-free. The objective is to quantify the number of bits required to achieve a desired competitive ratio (relative to an offline optimal algorithm). Examples include graph exploration [25], in which the explorer must visit all vertices in an unknown graph and return to its origin; tree exploration [27] in which the environment is a tree graph; exploration under different classes of advice oracles [31], in which the advice may encode aspects of the environment such as a port-numbered map of the graph or the starting point of the explorer, and treasure hunt [37, 49], in which a mobile searcher must locate a hider in an unknown environment. These works provided powerful information-theoretic lower bounds on the interplay between the size of advice and the competitive ratio, as well as explicit upper bounds via algorithms that leverage the corresponding advice.

Our work differs from the above studies in two important ways. First, unlike all previous works related to advice-complexity aspects of searching, we do not assume that the advice is perfect and error-free. Instead, we allow the possibility of worst-case manipulation of advice by an adversary, and provide results that explicitly quantify the effect of such manipulation, both from the upper and the lower bound point of view. Second, beyond the performance of the search strategies at the two extreme situations (precise versus adversarial advice), we provide a more nuanced analysis that expresses the competitive ratio as a function of the advice error, which is likewise novel in the analysis of search-related algorithms.

Furthermore, this work puts emphasis on the *semantic* information of hints. For positional and directional hints, this semantic information is straightforward: the hint provides the location or the direction of the target. But even for the setting of k -bit hints, our upper bounds interpret each bit as information with intuitive meaning. As we discuss in Section 5, each bit can be interpreted as a response to a *binary query*, which asks whether the target lies in some partition of the infinite line. Coupled with the error-based analysis, the upper bounds express the power of using k *imperfect* binary experts towards locating a hidden target.

Last, there is related work in which the searcher relies on advice in the form of noisy queries [19]. Specifically, the response to each query is correct with some known probability. In our setting, in contrast, we do not rely on any probabilistic assumptions in regards to the quality of the advice.

2. Preliminaries

In the context of searching on the line, a search strategy X can be defined as an infinite set of pairs (x_i, s_i) , with $i \in \mathbb{N}$, $x_i \in \mathbb{R}_{\geq 1}$ and $s_i \in \{0, 1\}$. We call i an *iteration* and x_i the *length* of the i -th search *segment*. More precisely, in the i -th iteration, the searcher

starts from the root O , traverses branch $s_i \bmod 2$ up to distance x_i from O , then returns to O . It suffices to focus on strategies for which $x_{i+2} \geq x_i$, i.e., in any iteration the searcher always searches a new part of the line. We will sometimes omit the s_i 's from the definition of the strategy, if the direction is not important, i.e., the searcher can start by moving either to the left or to the right of O . In this case, there is the implicit assumption that s_i and s_{i+1} have complementary parities, since any strategy that revisits the same branch in consecutive iterations can be transformed to another strategy that is no worse, and upholds the assumption. We make the standing assumption that the target lies within distance at least a fixed value, otherwise every strategy has unbounded competitive ratio. In particular, we will assume that t is such that $d(t) \geq 1$.

Given a strategy $X = (x_0, x_1, \dots)$ (which we will denote by $X = (x_i)$, for brevity). It is known that its competitive ratio is maximized for targets hiding immediately after the turn point of each segment (see e.g., [51]). Consider a target hiding at the end of segment $i - 1$, i.e., at distance infinitesimally larger than x_{i-1} . In this case, the strategy locates the target t at cost $x_{i-1} + 2 \sum_{j=1}^i x_j$, whereas $d(t) = x_{i-1}$. Hence,

$$\text{cr}(X) = 1 + 2 \sup_{i \geq 0} \frac{\sum_{j=1}^i x_j}{x_{i-1}}, \quad (3)$$

where x_{-1} is defined to be equal to 1.

Geometric sequences are important in search problems, since they often lead to efficient, or optimal strategies (see, e.g., Chapters 7 and 9 in [1]). We call the search strategy $G_b = (b^i)$ *geometric with base b* . From (3), we obtain that

$$\text{cr}(G_b) = 1 + 2 \frac{b^2}{b-1}. \quad (4)$$

For example, for the standard doubling strategy in which $x_i = 2^i$, hence $b = 2$, the above expression implies a competitive ratio of 9.

For any $r \geq 9$ define ρ_r to be such that $r = 1 + 2\rho_r$, thus $\rho_r = (r - 1)/2$. From (4), it follows that for the geometric strategy G_b to be r -competitive, it suffices that $1 + 2 \frac{b^2}{b-1} \leq r$, or equivalently, $\frac{b^2}{b-1} \leq \rho_r$. Note that the function $x^2/(x-1) - \rho_r$ has two roots, the largest of which is given by (2), and will be denoted by b_r . This expression will be prominent in the analysis of our strategies, and we will use the following fact:

Remark 1. *For the geometric strategy $G(b_r)$ with base b_r , it holds that $\text{cr}(G_{b_r}) = r$.*

In the context of searching with a hint, we will say that a strategy is (c, r) -competitive if it has consistency at most c and robustness at most r ; equivalently we say that the strategy is c -consistent and r -robust. Clearly, an r -robust strategy gives rise to a strategy with no hints, and with competitive ratio at most r .

We conclude with some definitions that will be useful in Section 4. Let $X = (x_0, x_1, \dots)$ denote a sequence of positive numbers. We define α_X as

$$\alpha_X = \overline{\lim}_{n \rightarrow \infty} x_n^{1/n}.$$

We also define as X^{+i} the subsequence of X starting at i , i.e. $X^{+i} = (x_i, x_{i+1}, \dots)$. Last, we define the sequence $G_b(\gamma_0, \dots, \gamma_{n-1})$ as

$$G_b(\gamma_0, \dots, \gamma_{n-1}) = (\gamma_0, \gamma_1 a, \gamma_2 a^2, \dots, \gamma_{n-1} a^{n-1}, \gamma_0 a^n, \gamma_1 a^{n+1}, \dots).$$

3. Hint is the position of the target

In this section we study the setting in which the hint is related to the exact position of the target. Namely, the hint h describes the distance $d(t)$ of the target t from the root, as well as the branch on which it hides. For any $r \geq 9$, we will give a strategy that is $(\frac{b_r+1}{b_r-1}, r)$ -competitive. Moreover, we will show that this is Pareto-optimal. We begin with the upper bound.

Theorem 2. *For any $r \geq 9$ there exists a $(\frac{b_r+1}{b_r-1}, r)$ -competitive strategy for linear search in which the hint is the position of the target.*

Proof. From the hint h , we have as information the distance $d(t)$ as well as the branch on which the target t lies; without loss of generality, suppose that this branch is the branch 0. Recall that this information may or may not be correct, and the searcher is oblivious to this.

Consider the geometric strategy $G_{b_r} = (b_r^i)$, with $i \in \mathbb{N}$, and recall that G_{b_r} is r -robust (Remark 2). There must exist an index j_t such that $b_r^{j_t-2} < d(t) \leq b_r^{j_t}$. Define $\lambda = b_r^{j_t}/d(t) \geq 1$, and let G' denote the strategy $G' = (\{\frac{1}{\lambda} b_r^i, s_i\})$, where the s_i 's are defined such that that $s_{i+1} \neq s_i$, for all i , and $s_{j_t} = 0$.

In words, G' is obtained by “shrinking” the search lengths of G_{b_r} by a factor equal to λ , and by choosing the right parity of branch for starting the search, in a way that, if the hint is trusted, then in G' the searcher will locate the target right as it is about to turn back to O at the end of the j_t -th iteration.

Since G_{b_r} is r -robust, so is the scaled-down strategy G' . This follows immediately from the expression of the worst-case competitive ratio, namely (3). It remains then to bound the consistency $c_{G'}$ of G' . Suppose that the hint is trusted. We have that

$$d(G', t) = \frac{1}{\lambda} (2 \sum_{i=0}^{j_t-1} b_r^i + b_r^{j_t}),$$

and since $d(t) = b_r^{j_t}/\lambda$ we can bound $c_{G'}$ from above by

$$\frac{d(G', t)}{d(t)} = 1 + 2 \frac{b_r^{j_t} - 1}{b_r^{j_t} (b_r - 1)} \leq 1 + \frac{2}{b_r - 1} = \frac{b_r + 1}{b_r - 1}.$$

We conclude that G' is $(\frac{b_r+1}{b_r-1}, r)$ -competitive. □

Example 3. *For $r = 9$, Theorem 2 shows a 9-robust strategy that has consistency 3. In this strategy, a third of the search cost is dedicated to reaching the target in the last iteration, and two thirds of the cost are dedicated towards the exploration of the line in both directions, prior to finding the target.*

Next, we will show that the strategy of Theorem 2 is Pareto-optimal. To this end, we will need a technical lemma concerning the segment lengths of any r -robust strategy.

Lemma 4. *For any r -robust strategy $X = (x_i)$, it holds that*

$$x_i \leq \left(b_r + \frac{b_r}{i+1}\right)x_{i-1},$$

for all $i \geq 1$, where x_{-1} is defined to be equal to 1.

Proof. The proof is by induction on i . We first show the claim for $i = 0$. There exists a branch such that if a target t hides on it, and at distance $d(t) = 1 + \epsilon$, with ϵ infinitesimally small, then X locates t at cost $1 + 2x_0 + \epsilon$ (namely, t hides on the opposite branch explored by X in iteration 0). For $\epsilon \rightarrow 0$, and since X is r -robust, it must be that

$$1 + 2x_0 \leq r = 1 + 2\frac{b_r^2}{b_r - 1} \Rightarrow x_0 \leq \frac{b_r^2}{b_r - 1} \leq 2b_r,$$

where the last inequality follows from the fact that $b_r \geq 2$. Thus, the base case holds.

For the induction hypothesis, suppose that the claim holds for all $j \leq i$, that is $x_j \leq \left(b_r + \frac{b_r}{j+1}\right)x_{j-1}$, for all $j \leq i$. This implies that

$$x_{i-j} \geq \frac{1}{\prod_{k=0}^{j-1} \left(b_r + \frac{b_r}{i+1-k}\right)} x_i. \quad (5)$$

We will show that the claim holds for $i+1$. Consider a target t hiding at distance $d(t) = x_i + \epsilon$, with $\epsilon \rightarrow 0$, and at the same branch as the one explored by X in iteration i . Then X finds t at cost $d(t) + 2\sum_{j=0}^i x_j$, and since X is r -robust, from the definition of ρ_r in (2), it follows that

$$\frac{\sum_{j=0}^{i+1} x_j}{x_i} \leq \rho_r \Rightarrow x_{i+1} + \sum_{j=0}^{i-1} x_j \leq (\rho_r - 1)x_i,$$

and substituting x_0, \dots, x_{i-1} using (5), we obtain that

$$x_{i+1} \leq (\rho_r - 1 - P_i)x_i, \quad \text{where } P_i = \sum_{j=0}^{i-1} \frac{1}{\prod_{k=0}^{j-1} \left(b_r + \frac{b_r}{i+1-k}\right)}.$$

It then suffices to show that

$$\rho_r - 1 - P_i \leq b_r + \frac{b_r}{i+2} \quad \text{or equivalently } P_i \geq \rho_r - 1 - b_r \frac{i+3}{i+2}. \quad (6)$$

We will prove (6) by induction on i . For $i = -1$, (6) is equivalent to

$$2b_r \geq \frac{b_r^2}{b_r - 1} - 1,$$

which can be readily verified from the fact that $b_r \geq 2$. Assuming then that (6) holds for i , we will show that it holds for $i + 1$. We have

$$\begin{aligned}
P_{i+1} &= \frac{1}{b_r + \frac{b_r}{i+2}}(1 + P_i) && \text{(From the definition of } P_i) \\
&\geq \frac{1}{b_r + \frac{b_r}{i+2}}(1 + \rho_r - 1 - b_r \frac{i+3}{i+2}) && \text{(From induction hypothesis)} \\
&= \frac{i+2}{b_r(i+3)} \left(\frac{b_r^2}{b_r-1} - b_r \frac{i+3}{i+2} \right) && \text{(Since } \frac{b_r^2}{b_r-1} = \rho_r) \\
&> \frac{i+2}{i+3} \frac{b_r}{b_r-1} - 1.
\end{aligned}$$

To complete the proof of this lemma, it remains to show that

$$\frac{i+2}{i+3} \frac{b_r}{b_r-1} - 1 \geq \rho_r - 1 - b_r \frac{i+3}{i+2},$$

or equivalently, by substituting ρ_r with the expression $\frac{b_r^2}{b_r-1}$, that

$$\frac{i+2}{i+3} \frac{1}{b_r-1} + \frac{i+3}{i+2} \geq \frac{b_r}{b_r-1}.$$

The lhs of the above expression is decreasing in i , for every $b_r \geq 2$, thus the lhs is at least

$$\lim_{i \rightarrow \infty} \left(\frac{i+2}{i+3} \frac{1}{b_r-1} + \frac{i+3}{i+2} \right) = \frac{b_r}{b_r-1},$$

which concludes the proof. \square

We obtain a useful corollary concerning the sum of the first $i - 1$ search lengths of an r -robust strategy.

Corollary 5. *For any r -robust strategy $X = (x_i)$, it holds that*

$$\sum_{j=0}^{i-1} x_j \geq \frac{x_i}{1 + \frac{1}{i+1}} \left(\frac{b_r}{b_r-1} - \frac{i+2}{i+1} \right),$$

and for every $\epsilon \in (0, 1]$, there exists i_0 such that for all $i > i_0$, $\sum_{j=0}^{i-1} x_j \geq (\frac{1}{b_r-1} - \epsilon)x_i$.

Proof. We have

$$\begin{aligned}
\sum_{j=0}^{i-1} x_j &= x_{i-1} + \sum_{j=0}^{i-2} x_j \geq x_{i-1} \left(1 + \sum_{j=0}^{i-2} \frac{1}{\prod_{k=0}^{j-1} (b_r + \frac{b_r}{i+1-k})} \right) \\
&= x_{i-1} (1 + P_{i-1}) \geq x_{i-1} (\rho_r - b_r \frac{i+2}{i+1}) \\
&\geq \frac{x_i}{b_r + \frac{b_r}{i+1}} \left(\frac{b_r^2}{b_r-1} - b_r \frac{i+2}{i+1} \right),
\end{aligned}$$

where the first inequality follows from Lemma 4, the second inequality holds from the property on P_i that was shown in the proof of Lemma 4, and the last inequality follows again from Lemma 4.

We now observe that for sufficiently large i , the rhs of the inequality is arbitrarily close to $\frac{1}{b_r-1}x_i$, which concludes the proof. \square

We can now show a lower bound on the competitiveness of every strategy that matches the upper bound of Theorem 2.

Theorem 6. *For every (c, r) -competitive strategy for linear search in which the hint is the position of the target, it holds that $c \geq \frac{b_r+1}{b_r-1} - \epsilon$, for any $\epsilon > 0$.*

Proof. Let $X = (x_i)$ denote an r -robust strategy, with a hint that specifies the position of a target t . Suppose that X locates the target at the j_t -th iteration. We have that

$$c = \frac{d(X, t)}{d(t)} = \frac{2 \sum_{i=0}^{j_t-1} x_i + d(t)}{d(t)} \geq \frac{2 \sum_{i=0}^{j_t-1} x_i + x_{j_t}}{x_{j_t}} = 1 + 2 \frac{\sum_{i=0}^{j_t-1} x_i}{x_{j_t}}.$$

Note that the target t can be chosen to be arbitrarily far from O , which means that j_t can be unbounded (otherwise the strategy would not have bounded robustness). From Corollary 5 this implies that $\sum_{i=0}^{j_t-1} x_i$ can be arbitrarily close to $x_{j_t} \frac{1}{b_r-1}$, and therefore c is arbitrarily close to $1 + 2 \frac{1}{b_r-1} = \frac{b_r+1}{b_r-1}$, which concludes the proof. \square

4. Hint is the direction of search

In this section we study the setting in which the hint is related to the *direction* of the search. More precisely, the hint is a single bit that dictates whether the target is to the left or to the right of the root O . Again, we are interested in Pareto-optimal strategies with respect to competitiveness: namely, for any fixed $r \geq 9$, what is the smallest c such that there exist (c, r) -competitive strategies?

A related problem was studied by Schuierer [51], which is called *biased search*. One defines the *left* and *right* competitive ratios, as the competitive ratio of a search, assuming that the target hides to the left of the root, or to the right of the root, respectively. However, the searcher does not know the target's branch. Of course we know that the maximum of the left and the right competitive ratios is at least 9 (and for the doubling strategy, this is tight). [51] shows that for any search strategy on the line (not necessarily 9-robust), the *average* of the left and the right competitive ratios is at least 9. At first glance, one may think that this could be an unsurprising, and perhaps even trivial result; however this is not the case. The proof in [51] is not straightforward, and relies in a generalization of a theorem of [29] which lower bounds the supremum of a sequence of functionals by the supremum of much simpler, geometric functionals. We will discuss this theorem shortly.

The problem studied in [51] is related to our setting: the left and right competitive ratios correspond to the consistency c and the robustness r of the strategy. Hence from [51] we know that $c + r \geq 18$. However, there is a lot of room for improvement. In this section

we will show a much stronger tradeoff between c and r , and we will further prove that it is tight. For example, we will show that for any (c, r) -competitive strategy, if c approaches 5 from above, then r approaches infinity (in contrast, in this case, the lower bound of [51] yields $r \geq 13$). In fact, we will show that $c + r$ is minimized when $c = r = 9$. To this end, we will apply a parameterized analysis based on Schuierer's approach. We begin with the upper bound, by analyzing a specific strategy.

Theorem 7. *For every $b \geq 1$, and $\delta \in (0, 1]$, there is a (c, r) -competitive strategy for linear search with the hint being the direction of search, in which*

$$c = 1 + 2 \cdot \left(\frac{b^2}{b^2 - 1} + \delta \frac{b^3}{b^2 - 1} \right) \text{ and } r = 1 + 2 \cdot \left(\frac{b^2}{b^2 - 1} + \frac{1}{\delta} \frac{b^3}{b^2 - 1} \right).$$

Proof. Suppose, without loss of generality, that the hint points to branch 0. Consider a strategy $X = (\{x_i, i \bmod 2\})$, which starts with branch 0, and alternates between the two branches. This strategy has consistency and robustness given by the following expressions, as a consequence of (3):

$$c = 1 + 2 \cdot \sup_{k \geq 0} \left\{ \frac{\sum_{i=0}^{2k+1} x_i}{x_{2k}} \right\} \quad \text{and} \quad r = 1 + 2 \cdot \sup_{k \geq 0} \left\{ \frac{\sum_{i=0}^{2k} x_i}{x_{2k-1}} \right\}, \quad (7)$$

where x_{-1} is defined to be equal to 1.

In addition, the search lengths of X are defined by

$$x_i = b^i, \text{ if } i \text{ even and } x_i = \delta b^i, \text{ if } i \text{ is odd,}$$

where we require that $b > 1$, and $\delta \in (0, 1]$. Note that X is "biased" with respect to branch 0, which makes sense since the hint points to that branch.

Substituting these values into (7), we obtain that

$$\begin{aligned} c &= 1 + 2 \cdot \sup_{k \geq 0} \left\{ \frac{\sum_{i=0}^k b^{2i}}{b^{2k}} + \delta \frac{\sum_{i=0}^k b^{2i+1}}{b^{2k}} \right\} = 1 + 2 \cdot \sup_{k \geq 0} \left\{ \frac{b^{2(k+1)} - 1}{(b^2 - 1)b^{2k}} + \delta \frac{b^{2k+3} - 1}{(b^2 - 1)b^{2k}} \right\} \\ &\leq 1 + 2 \cdot \left(\frac{b^2}{b^2 - 1} + \delta \frac{b^3}{b^2 - 1} \right). \end{aligned}$$

Similarly, we have that

$$\begin{aligned} r &= 1 + 2 \cdot \sup_{k \geq 0} \left\{ \frac{1}{\delta} \frac{\sum_{i=0}^k b^{2i}}{b^{2k-1}} + \frac{\sum_{i=0}^{k-1} b^{2i+1}}{b^{2k-1}} \right\} = 1 + 2 \cdot \sup_{k \geq 0} \left\{ \frac{1}{\delta} \frac{b^{2(k+1)} - 1}{(b^2 - 1)b^{2k-1}} + \frac{b^{2k+1} - 1}{(b^2 - 1)b^{2k-1}} \right\} \\ &\leq 1 + 2 \cdot \left(\frac{1}{\delta} \frac{b^3}{b^2 - 1} + \frac{b^2}{b^2 - 1} \right). \end{aligned}$$

□

For example, if $\delta = 1$, and $b = 2$, then Theorem 7 shows that there exists a (9, 9)-competitive strategy. Interestingly, the theorem shows that as the consistency c approaches 5 from above, the robustness r of the strategy must approach infinity. This is because the function $\frac{b^2}{b-1}$ is minimized for $b = 2$, and hence for c to approach 5 from above, it must be that b approaches 2, and δ approaches 0. But then $\frac{1}{\delta}$ must approach infinity, and so must r .

We will show that the strategy of Theorem 7 is Pareto-optimal. To this end, we will use the following theorem of [51]. Recall the definitions of α_X , X^{+i} and $G_a(\gamma_0, \dots, \gamma_{n-1})$ given in Section 2.

Theorem 8 (Theorem 1 in [51]). *Let p, q be two positive integers, and $X = (x_0, x_1, \dots)$ a sequence of positive numbers with $\sup_{n \geq 0} x_{n+1}/x_n < \infty$ and $\alpha_X > 0$. Suppose that F_k is a sequence of functionals that satisfy the following properties:*

- (1) $F_k(X)$ depends only on $x_0, x_1, \dots, x_{pk+q}$,
- (2) $F_k(X)$ is continuous in every variable, for all positive sequences X ,
- (3) $F_k(aX) = F_k(X)$, for all $a > 0$,
- (4) $F_k(X + Y) \leq \max(F_k(X), F_k(Y))$, for all positive sequences X, Y , and
- (5) $F_{k+i}(X) \geq F_k(X^{+ip})$, for all $i \geq 1$.

Then there exist p positive numbers $\gamma_0, \gamma_1, \dots, \gamma_{p-1}$ such that

$$\sup_{0 \leq k < \infty} F_k(X) \geq \sup_{0 \leq k < \infty} F_k(G_{\alpha_X}(\gamma_0, \dots, \gamma_{p-1})).$$

We will use Theorem 8 to prove a tight lower bound on the competitiveness of any strategy X .

Theorem 9. *For every (c, r) -competitive strategy, there exists $\alpha > 1$, and $\delta \in (0, 1]$ such that $c = 1 + 2 \cdot (\frac{\alpha^2}{\alpha^2-1} + \delta \frac{\alpha^3}{\alpha^2-1})$, and $r = 1 + 2 \cdot (\frac{\alpha^2}{\alpha^2-1} + \frac{1}{\delta} \frac{\alpha^3}{\alpha^2-1})$.*

Proof. Let $X = (x_0, x_1, \dots)$ denote a (c, r) -competitive strategy, and suppose, without loss of generality, that the hint specifies that the target is in the branch labeled 0. There are two cases concerning X : either the first exploration is on the branch labeled 0, or on the branch labeled 1. Let us assume the first case; at the end, we will argue that the second case follows from a symmetrical argument. As we argued in the proof of Theorem 7, in this case the competitiveness of X is described by (7). Let us define the functionals

$$C_k = \frac{\sum_{i=0}^{2k+1} x_i}{x_{2k}} \quad \text{and} \quad R_k = \frac{\sum_{i=0}^{2k} x_i}{x_{2k-1}}.$$

Then we have that

$$c = 1 + 2 \cdot \sup_{k \geq 0} C_k \quad \text{and} \quad r = 1 + 2 \cdot \sup_{k \geq 0} R_k. \tag{8}$$

The functional C_k satisfies the conditions of Theorem 8 with $p = 2$, as shown in [51] therefore there exist $\gamma_0, \gamma_1 > 0$ such that

$$\begin{aligned} \sup_{k \geq 0} C_k &\geq \sup_{k \geq 0} C_k(G_{\alpha_X}(\gamma_0, \gamma_1)) = \sup_{k \geq 0} \frac{\gamma_0 + \gamma_1 \alpha_X + \gamma_0 \alpha_X^2 + \dots + \gamma_0 \alpha_X^2 k + \gamma_1 \alpha_X^{2k+1}}{\gamma_0 \alpha_X^{2k}} \\ &= \sup_{k \geq 0} \left\{ \frac{\sum_{i=0}^k \alpha_X^{2i}}{\alpha_X^{2k}} + \frac{\gamma_1 \sum_{i=0}^k \alpha_X^{2i+1}}{\gamma_0 \alpha_X^{2k}} \right\}. \end{aligned}$$

If $\alpha_X \leq 1$, then the above implies that $\sup_{k \geq 0} C_k = \infty$ (another way of dismissing this case is that if $\alpha_X \leq 1$, then X is bounded and the two branches are not explored to infinity, as required by any strategy of bounded consistency). We can thus assume that $\alpha_X > 1$, and we obtain that

$$\sup_{k \geq 0} C_k \geq \sup_{k \geq 0} \left\{ \frac{\alpha_X^{2k+2} - 1}{(\alpha_X^2 - 1) \alpha_X^{2k}} + \frac{\gamma_1}{\gamma_0} \frac{\alpha_X^{2k+3} - 1}{(\alpha_X^2 - 1) \alpha_X^{2k}} \right\} = \frac{\alpha_X^2}{\alpha_X^2 - 1} + \frac{\gamma_1}{\gamma_0} \frac{\alpha_X^3}{\alpha_X^2 - 1}. \quad (9)$$

We can lower-bound r using a similar argument. Namely, the functional R_k satisfies the conditions of Theorem 8, again as shown in [51] therefore

$$\begin{aligned} \sup_{k \geq 0} R_k &\geq \sup_{k \geq 0} R_k(G_{\alpha_X}(\gamma_0, \gamma_1)) = \sup_{k \geq 0} \frac{\gamma_0 + \gamma_1 \alpha_X + \gamma_0 \alpha_X^2 + \dots + \gamma_0 \alpha_X^{2k}}{\gamma_1 \alpha_X^{2k-1}} \\ &= \sup_{k \geq 0} \left\{ \frac{\gamma_0 \sum_{i=0}^k \alpha_X^{2i}}{\gamma_1 \alpha_X^{2k-1}} + \frac{\sum_{i=0}^k \alpha_X^{2i-1}}{\alpha_X^{2k-1}} \right\}. \end{aligned}$$

Using the same argument as earlier, it suffices to consider only the case $\alpha_X > 1$, in which case we further obtain that

$$\sup_{k \geq 0} R_k \geq \sup_{k \geq 0} \left\{ \frac{\gamma_0}{\gamma_1} \frac{\alpha_X^{2k+2} - 1}{(\alpha_X^2 - 1) \alpha_X^{2k-1}} + \frac{\alpha_X^{2k+1} - 1}{(\alpha_X^2 - 1) \alpha_X^{2k-1}} \right\} = \frac{\gamma_0}{\gamma_1} \frac{\alpha_X^3}{\alpha_X^2 - 1} + \frac{\alpha_X^2}{\alpha_X^2 - 1}.$$

We thus obtain that

$$\sup_{k \geq 0} R_k \geq \sup_{k \geq 0} \left\{ \frac{\gamma_0}{\gamma_1} \frac{\alpha_X^{2k+2} - 1}{(\alpha_X^2 - 1) \alpha_X^{2k-1}} + \frac{\alpha_X^{2k+1} - 1}{(\alpha_X^2 - 1) \alpha_X^{2k-1}} \right\} = \frac{\gamma_0}{\gamma_1} \frac{\alpha_X^3}{\alpha_X^2 - 1} + \frac{\alpha_X^2}{\alpha_X^2 - 1}. \quad (10)$$

Let us define $\delta = \frac{\gamma_1}{\gamma_0} > 0$. The result follows then by combining (8), (9) and (10). Note that if we require that $c \leq r$, it must be that $\delta \leq 1$, since $\alpha_X > 1$.

It remains to consider the symmetric case, in which in X , the first explored branch is branch 1. In this case the analysis is essentially identical: in (8) we substitute C_k with R_k and vice versa, in the expressions of c and r , and in the resulting lower bounds we require that $\delta > 1$. \square

It is important to note that in the proof of Theorem 9 we used the fact that the values γ_0 and γ_1 depend only on X and not on any functionals defined over X , as follows from the proof of Theorem 8 in [51].

Theorem 9 implies that any (c, r) -competitive strategy X is such that

$$c + r \geq 2 + 4 \frac{\alpha_X^2}{\alpha_X^2 - 1} + 2\left(\delta + \frac{1}{\delta}\right) \frac{\alpha_X^3}{\alpha_X^2 - 1},$$

which is minimized at $\delta = 1$, hence $c + r$ is minimized only if $c = r = 1 + 2 \frac{\alpha_X^2}{\alpha_X^2 - 1} \geq 9$. We conclude that the average of a strategy's consistency and robustness (or the average of the left and right competitive ratio, in the terminology of [51]) is minimized only by strategies that are 9-robust.

5. Hint is a k -bit string

In this section we study the setting in which the searcher has access to a *hint string* of k bits. We first consider the case $k = 1$. In Section 5.1 we will study the more general case.

It should be clear that even a single-bit hint is quite powerful, and that the setting is non-trivial. For example, the bit can indicate the right direction for search, as discussed in Section 4, but it allows for other possibilities, such as whether the target is at distance at most D from the root, for some chosen D . The latter was studied in [32], assuming that the hint is correct. More generally, the hint can induce a partition of the infinite line into two subsets L_1 and L_2 , such that the hint dictates whether the target is hiding on L_1 or L_2 .

We begin with the upper bound, namely we describe a specific search strategy, and the corresponding hint bit (as well as the query which it responds). Consider two strategies of the form

$$X_1 = (b_r^i) \text{ and } X_2 = (b_r^{i+\frac{1}{2}}).$$

Note that both X_1 and X_2 are r -robust: X_1 is geometric with base b_r , whereas X_2 is obtained from X_1 by scaling the search lengths by a factor equal to $b_r^{1/2}$. We also require that the two strategies start by searching the same branch, hence in every iteration, they likewise search the same branch.

We can now define a strategy Z with a single bit hint, which indicates whether the searcher should choose strategy X_1 or strategy X_2 . For any given target, one of the two strategies will outperform the other, assuming the hint is trusted. Thus, an equivalent interpretation of the hint is in the form of a partition of the infinite line into two sets L_1 and L_2 , such that if the target is in L_i , then X_i is the preferred strategy, with $i \in [1, 2]$. See Figure 1 for an illustration.

The following result bounds the performance of this strategy, and its proof will follow as a corollary of a more general theorem concerning k -bit strings that we show in Section 5.1 (Theorem 13).

Proposition 10. *For given $r \geq 9$, the above-defined strategy Z is r -robust and has consistency at most $1 + 2 \frac{a^{3/2}}{a-1}$, where $a = b_r$, if $r \leq 10$, and $a = 3$, otherwise.*

Note that if $r = 9$, then Z has consistency $1 + 4\sqrt{2} \approx 6.657$. For $r \in [9, 10]$, the consistency of Z is decreasing in r , as one expects. For $r \geq 10$, the consistency is $1 + 3\sqrt{3} \approx 6.196$.

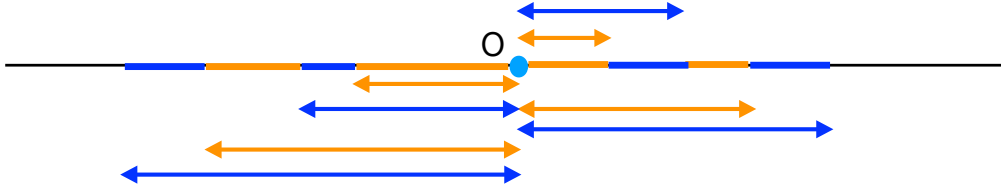


Figure 1: Illustration of strategy Z , using the first four segments of strategies X_1 and X_2 . Blue (dark) and orange (faded) segments correspond to the search segments of strategies X_1 and X_2 , respectively. The parts of the line in blue (resp. orange) indicate the hiding intervals for the target such that X_1 (resp. X_2) is preferred, and thus chosen by the hint.

We now turn our attention to lower bounds. To this end, we observe that a single-bit hint h has only the power to differentiate between two *fixed* strategies, say $X = (x_i)$, and $Y = (y_i)$, i.e., two strategies that are not defined as functions of h . We say that Z is *determined* by strategies X and Y , and the bit h .

Setting up the lower-bound proofs. We give some definitions and notation that will be used in the proofs of Theorems 11 and 12. Let Z be determined by strategies X and Y , and a single-bit hint h . Let C denote the lower bound on the consistency of Z that we wish to show. For given i , define $T_X^i = 2 \sum_{j=0}^i x_j + x_{i-1}$, and similarly for T_Y^i . Define also $q = r/C$.

Note that a searcher that follows strategy X will turn towards the root at iteration $i - 1$, after having explored some branch $\beta_i \in \{0, 1\}$ up to distance x_{i-1} . Thus, X barely misses a target that may be hiding at branch β_i , and at distance $x_{i-1} + \epsilon$ from O , with $\epsilon > 0$ infinitesimally small, and thus requires time T_X^i to discover it. We will denote this hiding position of a potential target by P_i . If, on the other hand, the searcher follows Y , then it can locate a target at position P_i at a time that may be smaller than T_X^i ; let τ_i denote this time. When strategy Y locates a target hiding at P_i , it does so by exploring branch β_i to a length greater than x_{i-1} . Let j_i be the iteration at which Y locates P_i , thus $y_{j_i} \geq x_{i-1}$. Last, let Q_i denote the position in branch β_i and at distance $y_{j_i} + \epsilon$ from O . In words, if a target hides at Q_i , then strategy Y barely misses it when executing the search segment y_{j_i} .

We first show a lower bound on the consistency of 9-robust strategies. In the proof we will not replace all parameters with the corresponding values (e.g., we will sometimes use r to refer to robustness, instead of the value 9). We do so because the arguments in the proof can be applied to other settings, as will become clear in the proof of Theorem 12.

Theorem 11. *For any $(c, 9)$ -competitive strategy with single-bit hint, it holds that $c \geq 5$.*

Proof. We will prove the result by way of contradiction. Let $C = 5$, and suppose that there is a strategy Z of consistency strictly less than C . Let Z be determined by two fixed strategies X and Y . Both X and Y must be r -robust (i.e., 9-robust), otherwise Z cannot be r -robust.

Fix $i_0 \in \mathbb{N}$. Suppose first that i_0 is such that for all $i \geq i_0$, we have $\tau_i \geq \frac{1}{q}T_X^i$. In this case, for a target at position P_i , defined earlier, X requires time T_X^i to locate it, whereas Y requires time at least $\tau_i \geq \frac{1}{q}T_X^i$ to locate it, thus the minimum time X or Y can locate this target is $\frac{1}{q}T_X^i$. Therefore, the consistency of Z is at least

$$c \geq \sup_{i \geq i_0} \frac{T_X^i}{q \cdot x_{i-1}} = \frac{1}{q} \sup_{i \geq i_0} \frac{T_X^i}{x_{i-1}} \geq \frac{C}{r} \cdot r = C, \quad (11)$$

which is a contradiction. Here, we used crucially the fact that $\sup_{i \geq i_0} \frac{T_X^i}{x_{i-1}} \geq 9$, for any 9-robust strategy X and any i_0 ¹. Specifically, there exists sufficiently large i such that T_X^i is arbitrarily close to $9x_{i-1}$.

It must then be that i_0 does not obey the property described above, namely for some $i \geq i_0$ we have that $\tau_i \leq \frac{1}{q}T_X^i$. Since X is $r = 9$ -robust, it must also be that $\frac{T_X^i}{x_{i-1}} \leq r$, as can be seen if a target hides at P_i . We therefore obtain that

$$\tau_i \leq \frac{1}{q}T_X^i \leq \frac{1}{q} \cdot r \cdot x_{i-1} = Cx_{i-1}. \quad (12)$$

We can also give a lower bound on τ_i , as follows. Recall that we denote by y_{j_i} the segment at which strategy Y locates a target at position P_i . For arbitrarily small $\epsilon > 0$, we can choose i_0 sufficiently large, which also implies that j_i can also be sufficiently large (since otherwise Y would not have finite robustness), so that Corollary 5 applies. To simplify the arguments, in the remainder of the proof we will assume that the corollary applies with $\epsilon = 0$; this has no effect on correctness, since we want to show a lower bound of the form $C - \delta$ on the consistency, and ϵ can be made as small as we want in comparison to δ . More precisely, we obtain that

$$\tau_i = 2 \sum_{l=0}^{j_i} y_l + x_{i-1} \geq \frac{2}{b_r - 1} y_{j_i} + x_{i-1}. \quad (13)$$

Combining (12) and (13) we have

$$y_{j_i} \leq \frac{C - 1}{2} (b_r - 1) x_{i-1}. \quad (14)$$

In particular, since $r = 9$ and $C = 5$, we have that $y_{j_i} \leq 2x_{i-1}$.

Consider now a target at position Q_i , and recall that this position is at distance infinitesimally larger than y_{j_i} . We will show that in both X and Y , there exists an $i \geq i_0$ such that the searcher walks distance at least $C \cdot y_{j_i}$ before reaching this position, which implies that Z has consistency at least C , and which yields the contradiction.

Consider first strategy Y . In this case, the searcher walks distance at least $T_Y^{j_i}$, to reach Q_i , from the definition of T_Y . Since $r = 9$, we know that $\sup_{i \geq i_0} \frac{T_Y^{j_i}}{y_{j_i}} \geq 9$, for any i_0 , hence

¹In general, this statement is not immediately true for arbitrary $r > 9$.

there exists an $i \geq i_0$ such that the distance walked by the searcher is at least $r \cdot y_{j_i}$, and hence at least $C \cdot y_{j_i}$.

Consider now strategy X . In this case, in order to arrive at position Q_i , the searcher needs to walk distance T_X^i , then at least an additional distance $y_{j_i} - x_{i-1}$ to reach Q_i . Let us denote by D_X^i this distance. We have

$$\begin{aligned} D_X^i &= T_X^i + y_{j_i} - x_{i-1} \\ &\geq 9x_{i-1} + y_{j_i} - x_{i-1} \quad (\text{From Corollary 5 and since } T_X^i \text{ is arbitrarily close to } 9x_{i-1}) \\ &= 8x_{i-1} + y_{j_i}. \end{aligned} \tag{15}$$

We then bound the ratio D_X^i/y_{j_i} from below as follows.

$$\begin{aligned} \frac{D_X^i}{y_{j_i}} &\geq \frac{8x_{i-1} + y_{j_i}}{y_{j_i}} = 1 + 8 \frac{x_{i-1}}{y_{j_i}} \\ &\geq 1 + 8 \frac{x_{i-1}}{2 \cdot x_{i-1}} \quad (\text{From the fact that } y_{j_i} \leq 2 \cdot x_{i-1}) \\ &= 5. \end{aligned} \tag{16}$$

We thus conclude that $C \geq 5$, which yields the contradiction, and completes the proof. \square

Showing a lower bound on the consistency, as a function of general $r > 9$ is quite hard, even for the case of a single-bit hint. The reason is that as r increases, so does the space of r -robust strategies. For example, any geometric strategy G_b has robustness r , as long as $b \in [\frac{\rho_r - \sqrt{\rho_r^2 - 4\rho_r}}{2}, \frac{\rho_r + \sqrt{\rho_r^2 - 4\rho_r}}{2}]$. In what follows we will show a lower bound for a class of strategies which we call *asymptotic*. More precisely, recall the definition of T_X^i . We call an r -robust strategy S *asymptotic* if $\sup_{i \geq i_0} \frac{T_X^i}{x_{i-1}} = r$, for all fixed i_0 . In words, in an asymptotic strategy, the worst-case robustness (i.e., the worst case competitive ratio without any hint) can always be attained by targets placements sufficiently far from the root. All geometric strategies, including the doubling strategy, have this property, and this holds for many strategies that solve search problems on the line and the star, such as the ones described in the introduction. Note also that the strategies X_1 and X_2 that determine the strategy Z in the statement of Proposition 10 are asymptotic, since they are near-geometric. Thus, the lower bound we show in the next theorem implies that in order to substantially improve consistency, one may have to resort to much more complex, and most likely irregular strategies.

Theorem 12. *Let Z denote a strategy with 1-bit hint which is determined by two r -robust, asymptotic strategies X and Y . Then Z is (c, r) -competitive, with $c \geq 1 + \frac{2b_r}{b_r - 1}$.*

Proof. We show how to modify the proof of Theorem 11. Let C be equal to $1 + \frac{2b_r}{b_r - 1}$, and suppose, by way of contradiction, that the robustness of Z is strictly less than C . First, we note that (11) applies since X is asymptotic, and so do equations (12), 13 and (14).

As in the proof of Theorem 11, we next consider a target hiding at position Q_i . We then can argue that there exists $i \geq i_0$ such that, for this hiding position, the total distance walked by Y is at least $r \cdot y_{j_i} \geq C y_{j_i}$. Here we use the fact that Y is asymptotic.

Next we consider strategy X , and we bound its cost for locating the hiding position Q_i . We have, similarly to the proof of Theorem 11, that

$$D_X^i = T_X^i + y_{j_i} - x_{i-1} \geq (r-1)x_{i-1} + y_{j_i}. \quad (\text{Since } X \text{ is asymptotic})$$

Therefore

$$R \geq \frac{D_X^i}{y_{j_i}} \geq 1 + \frac{(r-1)x_{i-1}}{y_{j-1}} \geq 1 + \frac{2(r-1)}{(C-1)(b_r-1)},$$

where the last inequality follows from (14). Solving this inequality for C we obtain

$$C \geq 1 + \sqrt{\frac{2(r-1)}{b_r-1}},$$

and substituting with $r = 1 + 2\frac{b_r^2}{b_r-1}$ we obtain that $C \geq 1 + \frac{2b_r}{b_r-1}$, a contradiction, and the proof is complete. \square

5.1. k -bit hints

Here we consider the general setting in which the hint is a k -bit string, for some fixed k . First, we give an upper bound that generalizes Proposition 10. Consider 2^k strategies X_0, \dots, X_{2^k-1} , where

$$X_j = (a^{i+\frac{j}{2^k}})_{i \geq 0},$$

for some a to be determined later, and where all the X_j have the same parity: they all search the same branch in their first iteration and, therefore in every iteration as well. Define a strategy Z , which is determined by X_0, \dots, X_{2^k-1} , and in which the k -bit hint h dictates the index j of the chosen strategy X_j . In other words, h answers the query $Q_h =$ “which strategy among X_0, \dots, X_{2^k-1} should the searcher choose?”. An equivalent interpretation is that the statements of the X_j ’s induce a partition of the line, such that for every given target position, one of the X_j ’s is the preferred strategy. Thus every bit i of the hint can be thought, equivalently, as the answer to a partition query Q_i of the line, i.e., of the form “does the target belong in a subset L_i of the line or not?”.

Theorem 13. *For every $r \geq 9$, the strategy Z defined above is (c, r) -competitive with $c \leq 1 + 2\frac{a^{1+\frac{1}{2^k}}}{a-1}$, where $a = b_r$, if $\rho_r \leq \frac{(1+2^k)^2}{2^k}$, and $a = 1 + 2^k$, otherwise.*

Proof. For a given target t , let $j_t = h$, that is, j_t is the index of the best strategy among the X_j ’s for located t , as dictated by h . From the statements of the strategies, this implies that there exists some i_t such that

$$a^{i_t+\frac{j_t}{2^k}-1-\frac{1}{2^k}} < d(t) \leq a^{i_t+\frac{j_t}{2^k}},$$

Then

$$\frac{d(X_{j_t}, t)}{d(t)} = \frac{2 \sum_{l=0}^{i_t-1} a^{l+\frac{j_t}{2^k}} + d(t)}{d(t)} \leq 1 + 2 \frac{\sum_{l=0}^{i_t-1} a^{l+\frac{j_t}{2^k}}}{a^{i_t+\frac{j_t}{2^k}-1-\frac{1}{2^k}}} \leq 1 + 2 \frac{a^{1+\frac{1}{2^k}}}{a-1}.$$

It remains to choose the right value for a . Recall that the geometric strategy G_a is $(1 + 2\frac{a^2}{a-1})$ -robust, thus so is every X_j defined earlier. We then require that a is such that

$$1 + 2\frac{a^2}{a-1} \leq r = 1 + 2\rho_r \Rightarrow \frac{a^2}{a-1} \leq \rho_r.$$

We also require that the consistency of Z , as bounded earlier, is minimized. Therefore, we seek $a > 1$ such that $\frac{a^{1+\frac{1}{2^k}}}{a-1}$ is minimized and $\frac{a^2}{a-1} \leq \rho_r$. Using simple calculus, we obtain that the value of a that satisfies the above constraints is as in the statement of the theorem. \square

For example, for $r = 9$, we obtain a strategy that is $(1 + 2^{2+\frac{1}{2^k}}, 9)$ -competitive. Thus, the consistency decreases rapidly, as function of k , and approaches 5.

Last, it is easy to see that no 9-robust strategy with hint string of any size can have consistency better than 3. To see this, let X be any 9-robust strategy, and let i_t be the iteration in which it locates a target t . Since t can be arbitrarily far from O , i_t is unbounded, and thus Corollary 5 applies. We thus have that

$$\frac{d(X, t)}{d(t)} = \frac{\sum_{j=0}^{i_t-1} x_j + d(t)}{d(t)} = 1 + 2 \frac{\sum_{j=0}^{i_t-1} x_j}{d(t)} \geq 1 + 2 \frac{\sum_{j=0}^{i_t-1} x_j}{x_{i_t}} \geq 1 + 2 \frac{(1-\epsilon)x_{i_t}}{x_{i_t}},$$

thus the consistency of X cannot be smaller than $3 - \epsilon$, for any ϵ . The same holds then for any strategy that is determined by any number of 9-robust strategies, and thus for any hint.

6. Dealing with errors

In all previous sections, we studied consistency/robustness tradeoffs in online search. In this section we will discuss the case in which the hint comes with some error η , as in recent works in Machine Learning [45, 50, 46]. The ultimate goal here is to describe Pareto-efficient solutions in which the competitive ratio of the search is also robust to the error. Note that the concept of error makes a lot of sense for positional hints (Section 3) or advice-string hints (Section 5); in contrast, for the directional hint we studied in Section 4, the Pareto-optimality results inherently captures the possibility of directional error.

Before we proceed, we first need to point out certain subtleties in regards to performance evaluation as function of the error, which are specific to search problems. That is, unlike other online problems, the competitive ratio of linear search is not necessarily an increasing function of the error η . For example, consider the case in which there is a prediction of the position of the target, and suppose that we have a “bad” search strategy (e.g, a strategy with very large search lengths, which results in a bad competitive ratio). Suppose that we are given a positional hint for a target, which is also very bad: the actual target hides very far away from the position of the hint. However, it is possible that this strategy performs

extremely well, if the target happens to hide right before the turn point of one of the search lengths. In this case, a bad strategy turns out to be quite efficient, even if the hint is also quite bad. These are complications that need to be taken into consideration in terms of defining efficiency in this setting, and highlight the difficulties in filtering out such outliers from the analysis, without resorting to some assumptions on the amount of the error.

6.1. Positional hints

Suppose that the searcher is given a hint h that describes the position of the target t on the line, and let $d(h)$ denote the distance of the hint from the origin. We will make the assumption that h and t are both on the same branch, and we define the error η to be such that either $d(h) = d(t)(1 - \eta)$, if $d(h) < d(t)$, and $d(h) = d(t)(1 + \eta)$, if $d(h) \geq d(t)$. In the former case, we call the error *negative*, whereas in the second case we call the error *positive*. This is only be for the purposes of the analysis, since the searcher does not know the parity of the error. Note that if the error is negative, then from the assumption it must be that $\eta \in [0, 1]$.

We discuss how to adapt the strategy of Section 3 to this setting. For given hint h , let X_h denote the Pareto-optimal strategy of Theorem 2. This strategy cannot tolerate any errors: if η is infinitesimally small positive error, then X_h barely misses the target in the crucial iteration, and the competitive ratio of the strategy becomes as bad as its worst-case robustness. This example shows that one needs to further “stretch” the search lengths of X_h , so as to robustify the strategy against such extreme situations. Let $p > 0$ denote a parameter that is specified by the searcher, and which captures the degree to which it is able to tolerate errors, and consider the strategy $X_{h(1+p)}$, i.e., the strategy that pretends that the positional hint is at distance $d(h)(1 + p)$ from O . The following is a corollary of Theorem 2.

Corollary 14. *For any $p \in [0, 1]$ and $r \geq 9$, strategy $X_{h(1+p)}$ is r -robust and has consistency at most $\min\{1 + 2\frac{1+p}{b_r-1}, r\}$. It also has competitive ratio at most $\min\{1 + 2\frac{1+p}{b_r-1}, r\}$, for any negative error, or positive error at most p , and at most r , in all other cases.*

6.2. k -bit hints

In this setting, we assume that a number $\eta \leq k$ of hint bits may be erroneous. As with positional hints, the searcher specifies a parameter $p \leq k$ that describes its desired tolerance to errors, or, alternatively, an anticipated upper bound on the error. We will first outline our approach. Let $\mathcal{X}_{a,2^k}$ denote the collection of the 2^k strategies in the proof of Theorem 13. We will show that this collection has a nice structural property (Property 15): given a target T , if j_T is the index of the strategy that locates T at minimum cost among the strategies in $\mathcal{X}_{a,2^k}$, then a strategy of index $j \in [j_T, 2^k - 1]$, will find T at cost at most a^{j-j_T} times the cost of strategy j_T . In the next step, we show how to apply a *fault-tolerant* binary-search algorithm, using as hint the response to k binary queries (in the presence of errors), so as to find, within the interval $[0, 2^k - 1]$, a strategy in $\mathcal{X}_{a,2^k}$ of index close (and exceeding) j_T . Last, in Theorem 16, we show how to evaluate the competitive ratio and optimize the choice of a .

Property 15. Consider the set $\mathcal{X}_{a,2^k} = \{X_0, \dots, X_{2^k-1}\}$ of 2^k single-searcher strategies, as defined in the proof of Theorem 13. For a given hiding target T , let $j_T \in [0, 2^k - 1]$ be such that X_{j_T} locates T at minimum cost among strategies in $\mathcal{X}_{a,2^k}$, and specifically at cost $c_0 + d(T)$, for some c_0 . Then, for all $j \in [j_T, 2^k - 1]$, X_j finds T at cost at most $a^{j-j_T} c_0 + d(T)$.

Proof. Follows straightforwardly from the definition of $\mathcal{X}_{a,2^k}$, and the observation that if X_{j_T} finds T in round i_T , then X_j , with $j \in [j_T, 2^k - 1]$ finds T no later than round i_T . \square

From Property 15, the searcher would like to leverage the hint so as to find the index of a strategy in $\mathcal{X}_{a,2^k}$ that is as close to j_T as possible, but is larger than j_T . This can be accomplished using the *Robust Binary Interval Search* (RBIS) algorithm of [24], which was developed in the context of an online problem known as *time-series search*, and which is based on an algorithm for a related fault-tolerant setting due to [24]. The same, essentially, algorithm, can be applied in our setting with only a minor modifications. Namely, we use the algorithm of [7] so as to search a binary tree that corresponds to the set of all strategy indices $\{0, \dots, 2^k - 1\}$, and knowing that the search ends in a node of height at most $\lfloor k/2 \rfloor + 2p$, we return the rightmost leaf of the subtree rooted at that node. Here, each one of the k bits corresponds to a query of the form “Is $j_T \leq x$?”, for some $x \in [0, 2^k - 1]$.

Combining Property 15, the fault-tolerant search algorithm RBIS, and an analysis along the lines of the proof of Theorem 13, we obtain the following result.

Theorem 16. For any $r \geq 9$, there exists a strategy with k -bit hint that has competitive ratio at most

$$1 + 2 \frac{a^{1+\frac{1+U}{2^k}}}{a-1}, \text{ where } a = \begin{cases} b_r, & \text{if } \rho_r \leq \frac{(1+2^k/(1+U))^2}{2^k/(1+U)} \\ 1 + 2^k/(1+U), & \text{otherwise,} \end{cases} \quad (17)$$

if $\eta \leq p \leq k/4$, where $U = 2^{\lfloor k/2 \rfloor + 2p}$. Otherwise, the competitive ratio of the strategy is at most r .

Proof. We apply RBIS on the set of indices of all strategies in $\mathcal{X}_{a,2^k}$, where $a > 1$ will be chosen later. Assuming that $\eta \leq p$, then the output is the index of a strategy in $\mathcal{X}_{a,2^k}$ which is ranked at most U among the strategies in $\mathcal{X}_{a,2^k}$, in regards to its cost for locating the target (where lower ranking indicates better performance). From Property 15, this means that the selected strategy has cost at most $a^U c_0 + d(t)$, where $c_0 + d(t)$ is the cost of the best strategy in $\mathcal{X}_{a,2^k}$. It is easy to evaluate the competitive ratio of this strategy along the lines of the proof of Theorem 13, and we infer that the competitive ratio of the selected strategy is at most $1 + 2 \frac{a^{2^k+1+U}}{a^{2^k}-1}$. The proof follows by choosing $a > 1$ that minimizes this expression. \square

We conclude with an observation about the performance of the strategy of Theorem 16. As k increases, and more specifically as $k \rightarrow \infty$, the competitive ratio of the strategy converges to that of Theorem 13, even if p is as large as the highest tolerance of RBIS to

errors, namely $k/4 - o(k)$. In other words, even if we allow error as high as $k/4 - o(k)$, the strategy of Theorem 16 remains very robust to errors, provided that the size of the hint is sufficiently large.

7. Conclusion

We introduced a model that allows us to analyze the performance of search strategies with some information on the target. As explained in the introduction, the eventual goal is to apply this model to many other search problems, within or beyond competitive analysis, and for deterministic or randomized strategies. A good starting point is the generalization of linear search, namely the m -ray star problem, which has a long history of research, and has a broad connection to resource allocation problems (see, e.g. [39, 8, 18]). In this context, there are m tasks and the goal is to allocate resources to each of them, without knowing ahead of time which task will be the crucial one. The hint can be interpreted as some information concerning the instance (e.g., the index of the crucial task, or the time we may have to allocate to each one of the tasks in order to guarantee some good solution). A more ambitious goal is to study searching in general graphs; here the hints could provide, for instance, some information about a neighborhood of nodes in which the target is likely to hide, or one may choose again to encode the hint as a k -bit advice string.

Funding Research benefited from the support of the FMJH Program PGMO and from the support to this program from EDF-THALES-ORANGE. Also partially funded by the grant ANR-19-CE48-0016 from the French National Research Agency (ANR).

Acknowledgements The author is thankful to Thomas Lidbetter for several comments on an earlier version of this paper.

References

- [1] Steve Alpern and Shmuel Gal. *The theory of search games and rendezvous*. Kluwer Academic Publishers, 2003.
- [2] 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.
- [3] Spyros Angelopoulos, Diogo Arsénio, and Christoph Dürr. Infinite linear programming and online searching with turn cost. *Theoretical Computer Science*, 670:11–22, 2017.
- [4] Spyros Angelopoulos, Christoph Dürr, and Shendan Jin. Best-of-two-worlds analysis of online search. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019*, volume 126 of *LIPICs*, pages 7:1–7:17, 2019.

- [5] Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc P. Renault. Online computation with untrusted advice. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020*, pages 52:1–52:15, 2020.
- [6] Spyros Angelopoulos, Christoph Dürr, and Thomas Lidbetter. The expanding search ratio of a graph. *Discrete Applied Mathematics*, 260:51–65, 2019.
- [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. AAAI Press, 2022.
- [8] Yossi Azar, Andrei Z. Broder, and Mark S. Manasse. On-line choice of on-line algorithms. In Vijaya Ramachandran, editor, *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas, USA*, pages 432–440. ACM/SIAM, 1993.
- [9] Ricardo A. Baeza-Yates, Joseph C. Culberson, and Gregory G.E. Rawlins. Searching in the plane. *Information and Computation*, 106:234–244, 1993.
- [10] Anatole Beck. On the linear search problem. *Naval Research Logistics*, 2:221–228, 1964.
- [11] Anatole Beck. More on the linear search problem. *Israel Journal of Mathematics*, 3(2):61–70, 1965.
- [12] Anatole Beck and Micah Beck. Son of the linear search problem. *Israel Journal of Mathematics*, 48(2-3):109–122, 1984.
- [13] Anatole Beck and Micah Beck. The linear search problem rides again. *Israel Journal of Mathematics*, 53(3):365–372, 1986.
- [14] Anatole Beck and Micah Beck. The revenge of the linear search problem. *SIAM journal on control and optimization*, 30(1):112–122, 1992.
- [15] Anatole Beck and D.J. Newman. Yet more on the linear search problem. *Israel Journal of Mathematics*, 8:419–429, 1970.
- [16] Anatole Beck and Peter Warren. The return of the linear search problem. *Israel Journal of Mathematics*, 14(2):169–183, 1973.
- [17] R. Bellman. An optimal search problem. *SIAM Review*, 5:274, 1963.
- [18] Daniel S. Bernstein, Lev Finkelstein, and Shlomo Zilberstein. Contract algorithms and robots on rays: unifying two scheduling problems. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1211–1217, 2003.
- [19] Lucas Boczkowski, Amos Korman, and Yoav Rodeh. Searching a tree with permanently noisy advice. In *ESA 2018-26th Annual European Symposium on Algorithms*, pages 1–32, 2018.

- [20] Prosenjit Bose, Jean-Lou De Carufel, and Stephane Durocher. Searching on a line: A complete characterization of the optimal solution. *Theoretical Computer Science*, 569:24–42, 2015.
- [21] Joan Boyar, Lene M. Favrholdt, Christian Kudahl, Kim S. Larsen, and Jesper W. Mikkelsen. Online algorithms with advice: A survey. *SIGACT News*, 47(3):93–129, 2016.
- [22] Jurek Czyzowicz, Konstantinos Georgiou, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, and Sunil M. Shende. Search on a line by byzantine robots. In *27th International Symposium on Algorithms and Computation, ISAAC 2016, December 12-14, 2016, Sydney, Australia*, volume 64 of *LIPICs*, pages 27:1–27:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.
- [23] Erik D. Demaine, Sándor Fekete, and Shmuel Gal. Online searching with turn cost. *Theoretical Computer Science*, 361:342–355, 2006.
- [24] Yann Disser and Stefan Kratsch. Robust and adaptive search. In *34th Symposium on Theoretical Aspects of Computer Science, STACS*, volume 66 of *LIPICs*, pages 26:1–26:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.
- [25] Stefan Dobrev, Rastislav Kráľovič, and Euripides Markou. Online graph exploration with advice. In *Structural Information and Communication Complexity: 19th International Colloquium, SIROCCO 2012*, pages 267–278. Springer, 2012.
- [26] Rudolf Fleischer, Tom Kamphans, Rolf Klein, Elmar Langetepe, and Gerhard Trippen. Competitive online approximation of the optimal search ratio. *SIAM Journal on Computing*, 38(3):881–898, 2008.
- [27] Pierre Fraigniaud, David Ilcinkas, and Andrzej Pelc. Tree exploration with advice. *Information and Computation*, 206(11):1276–1287, 2008.
- [28] Drew Fudenberg and Jean Tirole. *Game theory*. MIT press, 1991.
- [29] Shmuel Gal. A general search game. *Israel Journal of Mathematics*, 12:32–45, 1972.
- [30] Shmuel Gal. Minimax solutions for linear search problems. *SIAM Journal on Applied Mathematics*, 27:17–30, 1974.
- [31] Barun Gorain and Andrzej Pelc. Deterministic graph exploration with advice. *ACM Transactions on Algorithms*, 15(1):1–17, 2018.
- [32] Patrick Jaillet and Matthew Stafford. Online searching. *Operations Research*, 49:234–244, 1993.
- [33] Ming-Yang Kao and Michael Littman. Algorithms for informed cows. In *Proceedings of the AAAI 1997 Workshop on Online Search*, 1997.

- [34] Ming-Yang Kao, Yuan Ma, Michael Sipser, and Yiqun Yin. Optimal constructions of hybrid algorithms. *Journal of Algorithms*, 29(1):142–164, 1998.
- [35] Ming-Yang Kao, John H Reif, and Stephen R Tate. Searching in an unknown environment: an optimal randomized algorithm for the cow-path problem. *Information and Computation*, 131(1):63–80, 1996.
- [36] David G. Kirkpatrick. Hyperbolic dovetailing. In *Proceedings of the 17th European Symposium on Algorithms (ESA)*, pages 616–627, 2009.
- [37] Dennis Komm, Rastislav Kráľovič, Richard Kráľovič, and Jasmin Smula. Treasure hunt with advice. In *Structural Information and Communication Complexity: 22nd International Colloquium, SIROCCO 2015*, pages 328–341. Springer, 2015.
- [38] Elias Koutsoupias, Christos H. Papadimitriou, and Michalis Yannakakis. Searching a fixed graph. In *Proc. of the 23rd Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 280–289, 1996.
- [39] Andrey Kupavskii and Emo Welzl. Lower bounds for searching robots, some faulty. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 447–453, 2018.
- [40] Russell Lee, Jessica Maghakian, Mohammad H. Hajiesmaili, Jian Li, Ramesh K. Sitaraman, and Zhenhua Liu. Online peak-aware energy scheduling with untrusted advice. In *e-Energy '21: The Twelfth ACM International Conference on Future Energy Systems*, pages 107–123. ACM, 2021.
- [41] Russell Lee, Bo Sun, John Lui, and Mohammad Hajiesmaili. Pareto-optimal learning-augmented algorithms for online k-search problems. *arXiv preprint arXiv:2211.06567*, 2022.
- [42] 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.
- [43] Alejandro López-Ortiz and Sven Schuierer. The ultimate strategy to search on m rays? *Theoretical Computer Science*, 261(2):267–295, 2001.
- [44] Alejandro López-Ortiz and Sven Schuierer. On-line parallel heuristics, processor scheduling and robot searching under the competitive framework. *Theoretical Computer Science*, 310(1–3):527–537, 2004.
- [45] Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, pages 3302–3311, 2018.

- [46] Arya Mazumdar and Barna Saha. Clustering with noisy queries. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5788–5799. Curran Associates, Inc., 2017.
- [47] Andrew McGregor, Krzysztof Onak, and Rina Panigrahy. The oil searching problem. In *Proceedings of the 17th European Symposium on Algorithms (ESA)*, pages 504–515, 2009.
- [48] Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, page 646–662. Cambridge University Press, 2021.
- [49] Andrzej Pelc and Ram Narayan Yadav. Advice complexity of treasure hunt in geometric terrains. *Information and Computation*, 281:104705, 2021.
- [50] Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. In *Advances in Neural Information Processing Systems*, volume 31, pages 9661–9670, 2018.
- [51] Sven Schuierer. Lower bounds in online geometric searching. *Computational Geometry: Theory and Applications*, 18(1):37–53, 2001.
- [52] Sven Schuierer. A lower bound for randomized searching on m rays. In *Computer Science in Perspective*, pages 264–277, 2003.
- [53] Bo Sun, Russell Lee, Mohammad Hajiesmaili, Adam Wierman, and Danny Tsang. Pareto-optimal learning-augmented algorithms for online conversion problems. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.
- [54] 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.