



City Research Online

City St George's, University of London

Citation: Cabello, S. & Giannopoulos, P. (2025). Searching in Euclidean Spaces with Predictions. Approximation and Online Algorithms: 22nd International Workshop, WAOA 2024,, pp. 31-45. doi: 10.1007/978-3-031-81396-2 ISSN 0302-9743 doi: 10.1007/978-3-031-81396-2

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/33527/>

Link to published version: <https://doi.org/10.1007/978-3-031-81396-2>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

Searching in Euclidean Spaces with Predictions

Sergio Cabello^{1,2} and Panos Giannopoulos³

¹ Faculty of Mathematics and Physics, University of Ljubljana, Slovenia

² Institute of Mathematics, Physics and Mechanics, Slovenia

`sergio.cabello@fmf.uni-lj.si`

³ Department of Computer Science, City, University of London, UK

`panos.giannopoulos@city.ac.uk`

Abstract. We study the problem of searching for a target at some unknown location in $\mathbb{R}^d$ when additional information regarding the position of the target is available in the form of predictions. In our setting, predictions come as approximate distances to the target: for each point $p \in \mathbb{R}^d$ that the searcher visits, we obtain a value $\lambda(p)$ such that $|pt| \leq \lambda(p) \leq c \cdot |pt|$, where $c \geq 1$ is a fixed constant, t is the position of the target, and $|pt|$ is the Euclidean distance of p to t . The cost of the search is the length of the path followed by the searcher. Our main positive result is a strategy that achieves $(12c)^{d+1}$ -competitive ratio, even when the constant c is unknown. We also give a lower bound of roughly $(c/16)^{d-1}$ on the competitive ratio of any search strategy in $\mathbb{R}^d$.

Keywords: search games · predictions · Euclidean space

1 Introduction

The problem of searching for a target positioned at some unknown location in some region is a classic search game problem that has been well-studied in both fields of Computational Geometry and Operations Research. The problem comes in many different versions including, linear search (i.e., searching for a target on a line) [6, 7, 9, 15], searching in the plane [4, 8, 16, 20], searching in concurrent rays [4, 14, 15], and searching inside polygonal regions [21, 26]. The book by Alpern and Gal [1] provides an extensive overview of general search games, while Ghosh and Klein [17] survey search problems in planar domains.

The searcher starts from some given position, follows some path according to some strategy until the target, usually a point, is reached or detected, for some appropriate definition of “detection”. In 1-dimensional settings, e.g., an infinite line, one usually requires that the searcher passes through the target point. In the plane one usually requires that the searcher is within some distance of the target, or sees the target, if there are obstacles, or that the target lies on the segment connecting the searcher’s current and starting positions [16, 17, 23].

The cost of the search is the length of the path followed by the searcher and the objective is to find an efficient search strategy. Efficiency is usually measured by the *competitive ratio*, which, in this setting, is the ratio of the length of the

path of the searcher to the actual Euclidean distance of the starting position to the target.

In this work, we consider the problem of finding a target point $\mathbf{t}$ in Euclidean space $\mathbb{R}^d$ when additional information regarding the position of the target is available in the form of *predictions*. Here, the predictions are the approximate distance to $\mathbf{t}$ for all points visited during the search, e.g., a value between, say, $|p\mathbf{t}|$ and $2|p\mathbf{t}|$ for each point p visited. Such an estimate could be obtained for example in a scenario where one takes into account the strength of a signal broadcasted by the target.

Algorithms with predictions is a concept that has been introduced relatively recently; see the survey by Mitzenmacher and Vassilvitskii [25]. The general idea is that on top of the usual input data we are also given additional and possibly inaccurate (noisy) information, the prediction, that should assist the algorithm to be more effective. The improvement in performance depends on the accuracy of the prediction.

We continue this section with the problem setup, a summary of our contribution, a short discussion on our predictions model, and related work.

1.1 Problem setup

We consider the following search problem in $\mathbb{R}^d$. Assume that there is a fixed but unknown *target* point $\mathbf{t} \in \mathbb{R}^d$. Without loss of generality, we start the search at the origin, which we denote by $\mathbf{o}$. We want to find a curve γ that starts at $\mathbf{o}$ and ends at $\mathbf{t}$. The *cost* of the search is the Euclidean length of the curve γ .

As we search for the target, we have approximate information about the distance to it from each point that we have visited in so far. More precisely, we assume that there is a constant $c \geq 1$ and an unknown function

$$\lambda: \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0} \text{ such that } \forall p \in \mathbb{R}^d \quad |p\mathbf{t}| \leq \lambda(p) \leq c \cdot |p\mathbf{t}|. \quad (1)$$

We refer to such a function λ as a *c-prediction* for the target $\mathbf{t}$. The constant c is the *prediction factor* of λ . See Figure 1 for an example in $d = 1$. Note that for $c = 1$, the function λ gives the exact distance to the target.

For each point p along the search path we have traversed so far, we obtain the value $\lambda(p)$, and the search strategy decides how to continue the search depending on that information. We know when we have reached the target because $\lambda(p) = 0$ holds only when $p = \mathbf{t}$.

As it is common in search games, we are interested in the competitive ratio of the search strategy: how does the length of the search path compares to the straight-line distance from the origin to the target? To formalize this, for each target $\mathbf{t} \in \mathbb{R}^d$ and each constant $c \geq 1$, we consider the family $A(\mathbf{t}, c)$ of c -predictions for $\mathbf{t}$, that is, functions satisfying condition (1). The family of c -prediction functions is $\cup_{\mathbf{t} \in \mathbb{R}^d} A(\mathbf{t}, c)$.

A search strategy S is $\alpha = \alpha(S, c, d)$ *competitive* if for all $\mathbf{t} \in \mathbb{R}^d$ and all $\lambda \in A(\mathbf{t}, c)$, the length of the path defined by S to reach $\mathbf{t}$ from $\mathbf{o}$ is at most $\alpha|\mathbf{o}\mathbf{t}|$. Note that we have two possible regimes, depending on whether c is known or unknown to the search strategy.

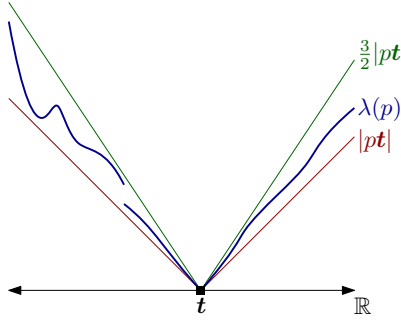


Fig. 1. Example of c -prediction function $\lambda(p)$ for $c = 3/2$. In this example the function is not monotone in $|pt|$ and it is not continuous.

1.2 Our contribution

Our main contributions are the following:

- We introduce a natural, new search problem in $\mathbb{R}^d$, under a predictions model where we have approximate information about the distance to the target.
- We show that for each dimension d and each constant prediction factor $c \geq 1$ there is a search strategy with competitive ratio $(6c)^{d+1}/3$. To achieve this, we use ε -nets from metric spaces, also known as r -nets, and provide a path of finite length but an infinite number of pieces. This result holds assuming that we know the prediction factor c . For *unknown* prediction factor, a similar statement holds with a slightly different search strategy and larger competitive ratio. This result is given in Section 3.
- We show that for $c > 4$, any deterministic search strategy in $\mathbb{R}^d$ with c -predictions will have a competitive ratio of at least $(1/4) \cdot (c/16)^{d-1} \cdot \min\{\sqrt{\pi/d}, 1\}$. For this result we again use ε -nets as a basic tool and we employ techniques developed for the Euclidean Traveling Salesperson with Neighbourhoods. With a slightly worse constant, the lower bound holds also for randomized search strategies. This lower bound is shown in Section 4.

Additionally, we give several basic properties of the model. For example, we show that without a prediction function throughout the whole search, we cannot find the target using a path of bounded length, and having an infinite number of pieces (segments) in the search path is unavoidable. We also show that we may assume that the prediction function is continuous. We also note that when $c = 1$, i.e., we have exact information about the distance to the target, the target can be reached with competitive ratio arbitrarily close to 1. A detailed discussion and proofs of these results can be found in the full version of the paper.

To our knowledge, this is the first search problem in Euclidean spaces with $d \geq 2$ where a search point reaches a target point with constant competitive ratio; see the discussion below about related work. For our upper and lower bounds we use bounds on the cardinality of ε -nets. For the sake of simplicity

in the calculations, we used suboptimal but simple-to-parse estimates in the simplifications and the cardinality of ε -nets. In any case, our upper and lower bounds for the best competitive ratio are still far apart.

Our model, as presented in the continuous setting, is scale-free and general. In the discrete setting, where the target has integral coordinates or where the target can be detected when the searcher is within a given distance from it, our strategies can be easily modified to have a finite number of steps. The lower bound holds in the discrete setting too, albeit with a slightly worse constant. Moreover, by having an approximate distance to the target as the prediction, the model can also accommodate underestimates of the actual distance (in addition to the default overestimates), which can be of interest in practical scenarios. These as well as other extensions are discussed in Section 3.1.

1.3 Related work

For *linear* search, when the exact distance to the target is known, one can easily find the target by walking at most three times this distance. When the distance to the target is unknown, Beck and Newman [7] and later Baeza-Yates et al. [4] showed that a simple doubling strategy has competitive ratio 9; here, a lower bound on the distance to the target is assumed, otherwise there is no search strategy with constant competitive ratio. There are also other similar strategies with the same competitive ratio [10, 17]. Moreover, various approaches [4, 7, 19, 22] show that 9 is the best possible competitive ratio for the problem.

Gal [14, 15] introduced the problem of searching for a target on multiple *rays* that are concurrent at the starting position and gave an optimal strategy for the case where the distance to the target is unknown. This result was rediscovered by Baeza-Yates et al. [4].

Baeza-Yates et al. [4] also considered the problem of finding a target with *integer coordinates* in the plane and presented various search strategies. When the distance to the target is known, it is also easy to get an optimal strategy. However, when the distance is unknown, and with no additional information available, no search strategy can have constant competitive ratio as there are $\Theta(n^2)$ integral points within distance at most n from the origin and any search strategy has to visit all of them in some order. For the natural extension of the problem in $\mathbb{R}^d$, the latter generalizes to any $d \geq 3$, as there are $\Theta(n^d)$ integral points within distance at most n from the origin. When we are in $\mathbb{R}^d$ and the distance to the target is known, we hit a classical problem in Number Theory: on how many ways can we express a positive integer as sum of d squares of integers. For $d \geq 5$, there is no search strategy with constant competitive ratio as there are $\Omega(n^{d/2-1})$ integral points at distance exactly n from the origin; see, for example, Vaughan and Wooley [27].

In another variant of the problem by Gal [16], the searcher travels along a path until the target lies on the *segment* connecting the searcher's current and starting positions, essentially sweeping around its starting position with an infinitely elastic cord until the target is swept. For this, Gal [16] gave a spiral

search strategy with 17.289 competitive ratio, which Langetepe [23] showed to be optimal.

Hipke et al. [19] considered linear search when the target is at *distance* at least 1 and at most $D \geq 1$ from the starting point, where D is known at the start of the search. Bose et al. [10] provided a more careful analysis using the roots of a recursive sequence of polynomials and gave better lower and upper bounds on the competitive ratio with dependence on D . López-Ortiz and Schuierer [24] considered also this setting, for the case of concurrent rays. Compared to these works, there are two main differences in our work. Firstly and most importantly, we have a prediction all the way through the search, while they have a prediction only at the start. Secondly, we consider the problem in more general settings, namely in $\mathbb{R}^d$ for arbitrary d . An upper bound at the start does not suffice to find a point when $d \geq 2$. (If $d = 2$ and the exact distance is known, the problem can be easily solved since the target has to lie on a known circle, and that is the only additional instance that is solvable.)

Banerjee et al. [5] considered the problem of finding a target in a *graph* with information about the distance to the target. In their model, the target is at one vertex of the graph and at each vertex we have a value stored that is made available only when we are adjacent to the vertex. For most vertices, the value stored at a vertex is the true distance from the vertex to the target, but for some vertices the value is wrong. Contrary to our setting, they do not assume a bounded error for the information at each vertex, but that the information is wrong at a bounded number of nodes. The bound on the number of nodes with wrong information then appears in the bound for the length of the search path.

Finally, Angelopoulos [2, 3] gave strategies for linear and multiple-ray search under a different model where a one-off, possibly erroneous hint or prediction on the target's position is given at the start of the search. The prediction can be positional, directional, or, in general, a k -bit string encoding answers to k binary queries and the measure of the performance of a strategy is a trade-off between the competitive ratio under error-free prediction and that under erroneous prediction.

2 Notation and preliminaries

Since the dimension d is always fixed and clear from the context, we drop in the notation the dependency on d .

The *ball* centered at $p \in \mathbb{R}^d$ with radius r is $B(p, r) = \{q \in \mathbb{R}^d \mid |pq| \leq r\}$. We will also consider the *spherical shells* $S(p, r_1, r_2) = \{q \in \mathbb{R}^d \mid r_1 \leq |pq| \leq r_2\}$. A spherical shell in the plane is an annulus. For a c -prediction function λ , whenever we are at a point $p \in \mathbb{R}^d$, we get a prediction $\lambda(p)$ and we deduce that the target point t lies in the spherical shell $S(p, \lambda(p)/c, \lambda(p))$. See Figure 2.

Note that we have made a modeling decision, namely we have assumed that there is an unknown function λ such that at each point p we get the prediction $\lambda(p)$. More generally, it could happen that we visit the same point p multiple times and at each time we get a different estimate of the distance from p to

the target. However, getting a different estimate can only help, as it provides more information: if we get two different c -predictions λ and λ' at different times at the same point p , then we know that the target lies in the spherical shell $S(p, \max\{\lambda, \lambda'\}/c, \min\{\lambda, \lambda'\})$. This is more information than what we get if $\lambda' = \lambda$ because then we can only conclude that the target lies in $S(p, \lambda/c, \lambda)$, which is strictly larger. Thus, when searching for an optimal search strategy, we can assume that each time we visit the same point we get the same prediction. In particular, a search strategy could simply ignore the information obtained in the second and subsequent visits to the same point.

A *path* in $\mathbb{R}^d$ is a continuous function $\pi: [0, 1] \rightarrow \mathbb{R}^d$. The paths in our search strategies will consist of an infinite number of straight-line segments and will exhibit a Zeno-like phenomena: they make an infinite number of turns in finite time and length. To show that the paths we define reach the target, we will use the following property, whose proof is an standard argument in continuity.

Lemma 1. *Let $\pi: [0, 1] \rightarrow \mathbb{R}^d$ be a path. Assume that there is a point $p \in \mathbb{R}^d$ with the following property: for each $\varepsilon > 0$ there exists some $\delta \in (0, 1]$ such that the subpath $\pi([1 - \delta, 1])$ is contained in $B(p, \varepsilon)$. Then $\pi(1) = p$, that is, p is the endpoint of the path π .*

We will use ε -nets from metric space theory: An *ε -net* for the ball $B(p, r)$ is a subset N of points from $B(p, r)$ such that (i) each point of $B(p, r)$ is at distance at most ε from some point of N , and (ii) each two distinct points of N are at distance at least ε . Condition (i) is equivalent to $B(p, r) \subseteq \bigcup_{q \in N} B(q, \varepsilon)$. Condition (ii) is equivalent to the balls $B(q, \varepsilon/2)$, where $q \in N$, being pairwise interior disjoint.

The following bound follows from a well-known technique using volumes.

Lemma 2. *In $\mathbb{R}^d$, for each $\varepsilon \leq r$, a ball of radius r has a ε -net with at least $(r/\varepsilon)^d$ and at most $(3r/\varepsilon)^d$ elements.*

3 Upper bound

In this section we provide search strategies to reach the target in $\mathbb{R}^d$ when we have a c -prediction. We first provide the key lemma that tells us how to get a sequence of points whose λ -values decreases geometrically. We then provide a strategy when the prediction factor is known, and then discuss how to handle the case for unknown prediction factor. In this setting, we adapt the notation so that c^* is the true prediction factor, while c is a guess for the true prediction factor. At the end of the section we discuss some extensions.

Lemma 3. *Assume that we are at point p_i and the prediction factor c^* is perhaps unknown. Let c be a guess for c^* . Using a search through a path γ_{i+1} of length at most $2(6c)^d \cdot \lambda(p_i)$ we get to one of the following outcomes:*

- we move from p_i to a point p_{i+1} such that $\lambda(p_{i+1}) \leq \lambda(p_i)/2$, or
- we come back to p_i and correctly deduce that $c < c^*$.

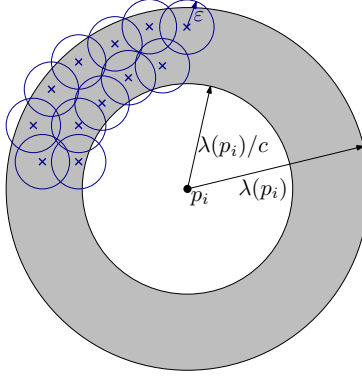


Fig. 2. Spherical shell $S(p_i, \lambda(p_i)/c, \lambda(p_i))$ where t must lie and (part of) a net.

Moreover, all points of the path γ_{i+1} are at distance at most $2\lambda(p_i)$ from t .

Proof. Recall that $|p_i t| \leq \lambda(p_i)$. Set $\varepsilon = \frac{\lambda(p_i)}{2c}$ and let N be a ε -net for the ball $B(p_i, \lambda(p_i))$ by Lemma 2. Since the target is contained in $B(p_i, \lambda(p_i))$ and N is a ε -net, there is some point $p_* \in N$ such that $|p_* t| \leq \varepsilon = \lambda(p_i)/2c$. If $c^* \leq c$, then we have $\lambda(p_*) \leq c^* \cdot |p_* t| \leq c^* \cdot \lambda(p_i)/2c \leq \lambda(p_i)/2$. If $c^* > c$, then we have no guaranteed useful bound for $\lambda(p_*)$.

We take a path that goes through the points of N in arbitrary order and, at each point of N , we query for the prediction $\lambda(\cdot)$. We finish the path as soon as we reach some point $q_* \in N$ such that $\lambda(q_*) \leq \lambda(p_i)/2$. If for all the points p of N we have $\lambda(p) > \lambda(p_i)/2$, then we go back to the point p_i . This finishes the description of the path γ_{i+1} .

In the first case, we set $p_{i+1} = q_*$ and we have moved to a point p_{i+1} with $\lambda(p_{i+1}) \leq \lambda(p_i)/2$. When $c^* \leq c$, we have to be in this case since the point p_* satisfies the stopping condition $\lambda(p_*) \leq \lambda(p_i)/2$. Thus, if we do not have such a point, we can conclude that $c^* > c$.

To bound the length of γ_{i+1} , we note that any two points in $B(p_i, \lambda(p_i))$ are at distance at most $2\lambda(p_i)$. We further note that the first edge has length at most $\lambda(p_i)$ and, if γ_{i+1} comes back to p_i , also the last edge has length at most $\lambda(p_i)$. Using the bound of Lemma 2 we get that the path γ_{i+1} has length at most

$$\lambda(p_i) + (|N| - 1) \cdot (2\lambda(p_i)) + \lambda(p_i) \leq \left(\frac{3\lambda(p_i)}{\varepsilon} \right)^d \cdot 2\lambda(p_i) = 2(6c)^d \cdot \lambda(p_i).$$

Finally, we note that the whole path γ_{i+1} is contained in $B(p_i, \lambda(p_i))$, which is contained in $B(t, 2\lambda(p_i))$ because $t \in B(p_i, \lambda(p_i))$.

Note: It is known [13] that for any set of n points in the d -dimensional ball $B(p, r)$ there is a tour of length $r \cdot O(n^{\frac{d-1}{d}})$ visiting them. This implies that we can also use a path γ_{i+1} of length $\lambda(p_i) \cdot O(|N|^{\frac{d-1}{d}}) = O((6c)^{d-1}) \cdot \lambda(p_i)$. With this, the dependency on c is slightly better at the expense of having more ugly-looking constants.

Theorem 1 (Known c). *Consider the search with predictions problem in $\mathbb{R}^d$ where the prediction factor $c^* > 1$ is known. There is a search strategy to reach the target with competitive ratio $2 \cdot 6^d \cdot (c^*)^{d+1}$.*

Proof. Let $p_0 = \mathbf{o}$ be the starting point and recall that $\lambda(p_0) \leq c^* \cdot |\mathbf{ot}|$.

For $i = 0, 1, 2 \dots$ iteratively, we use Lemma 3 with the guessed prediction factor $c = c^*$ to obtain a path γ_{i+1} . As the prediction factor is correct, we always have the outcome in the first item: γ_{i+1} finishes at a point p_{i+1} with $\lambda(p_{i+1}) \leq \lambda(p_i)/2$. It follows by induction that for each $i \in \mathbb{N}$ we have $\lambda(p_i) \leq \lambda(p_0)/2^i$ and therefore $\text{len}(\gamma_i) \leq 2(6c^*)^d \cdot \lambda(p_i) \leq 2(6c^*)^d \cdot \lambda(p_0)/2^i$.

Let γ be the concatenation of the paths $\gamma_1, \gamma_2, \dots$. Then

$$\begin{aligned} \text{len}(\gamma) &= \sum_{i=1}^{\infty} \text{len}(\gamma_i) \leq \sum_{i=1}^{\infty} 2(6c^*)^d \cdot \frac{\lambda(p_0)}{2^i} = 2(6c^*)^d \cdot \lambda(p_0) \\ &\leq 2 \cdot 6^d (c^*)^{d+1} \cdot |\mathbf{ot}|. \end{aligned}$$

The path makes an infinite number of straight-line moves. Since for each $i \in \mathbb{N}$, the suffix of the path γ after p_{i+1} is at distance at most $2 \cdot \lambda(p_{i+1}) \leq 2 \cdot \lambda(p_0)/2^{i+1} = \lambda(p_0)/2^i$ from $\mathbf{t}$, Lemma 1 implies that $\mathbf{t}$ is the endpoint of the path γ .

Theorem 2 (Unknown c). *Consider the search with predictions problem in $\mathbb{R}^d$ where the prediction factor c^* is unknown. There is a search strategy to reach the target with competitive ratio $(12c^*)^{d+1}$.*

Proof. The basic idea is using an exponential search for the constant c . At each step we use Lemma 3 to either move to a point with smaller predicted distance to the target or to detect that our guess for c is too small and double it. Index j parameterizes the current guess $c = 2^j$, and $p_i^{(j)}$ denotes a point during that guess. At each step we will increase either i or j .

We start setting $j = 1$, $i = 0$ and $p_i^{(j)} = p_0^{(1)} = \mathbf{o}$.

From the current point $p_i^{(j)}$, we use Lemma 3 with the guessed prediction factor $c = 2^j$ to obtain a path $\gamma_{i+1}^{(j)}$. If the outcome is given by the first item of Lemma 3, then we get to a point $p_{i+1}^{(j)}$ with $\lambda(p_{i+1}^{(j)}) \leq \lambda(p_i^{(j)})/2$; in this case we increase i . If the outcome is given by the second item of Lemma 3, then we get back to $p_i^{(j)}$; in this case we set $p_i^{(j+1)} = p_i^{(j)}$, increase j and, from this point on, we will use the new guessed prediction factor 2^j , which is twice larger than before. See Figure 3 for an schematic view of the sequence of points.

Let j^* be the largest value that index j takes through the procedure. Note that $j^* \leq \lceil \log_2 c^* \rceil$ since $2^{\lceil \log_2 c^* \rceil}$ is an overestimate to c^* . If we arrive to $j^* = \lceil \log_2 c^* \rceil$, from that point on we will always extend the search path using the outcome in the first item of Lemma 3. Set $J = \{1, \dots, j^*\}$, which is the set of values that index j takes through the procedure.

For each $j \in J$, let I_j be the set of indices i such that the point $p_i^{(j)}$ is defined and let $i_j = \max(I_j)$. Note that i_{j^*} is undefined because I_{j^*} is infinite, but i_j

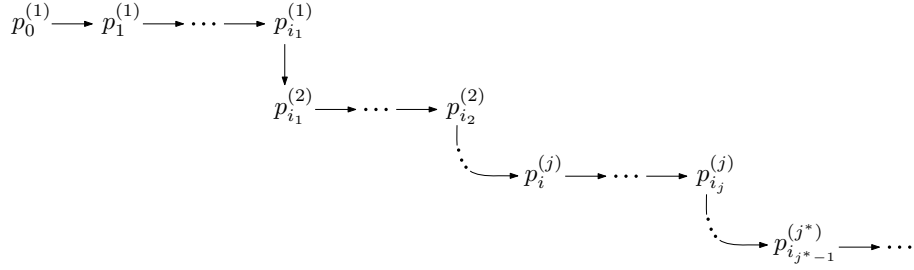


Fig. 3. Visualizing the sequence of points $\left((p_i^{(j)})_{i \in I_j} \right)_{j \in J}$. When increasing i we move right, when increasing j we move down. Here, we are using $j^* = \max(J)$ and $i_j = \max(I_j)$ for all $j \in J$.

is defined for all $j < j^*$ because I_j is finite for all $j < j^*$. By construction, the index i_j is the first element of I_{j+1} , for all $j < j^*$. It may happen that, for some $j < j^*$, the set I_j contains a single element. This happens when j is increased in successive steps and thus $p_i^{(j)} = p_i^{(j+1)} = p_i^{(j+2)}$.

It follows by induction that for all $j \in J$ and all $i \in I_j$ we have $\lambda(p_i^{(j)}) \leq \lambda(p_0^{(1)})/2^i = \lambda(\mathbf{o})/2^i$. Note that this bound is independent of the index j . Therefore

$$\forall j \in J, i \in I_j : \quad \text{len}(\gamma_{i+1}^{(j)}) \leq 2(6 \cdot 2^j)^d \cdot \lambda(p_i^{(j)}) \leq 2(6 \cdot 2^j)^d \cdot \frac{\lambda(\mathbf{o})}{2^i}.$$

Let γ be the concatenation of the paths $\gamma_{i+1}^{(j)}$, in the same order as we constructed them: first $\gamma_{i+1}^{(1)}$ for increasing $i \in I_1$, then $\gamma_{i+1}^{(2)}$ for increasing $i \in I_2$, and so on until we reach the infinite sequence $\gamma_{i+1}^{(j^*)}$ for increasing $i \in I_{j^*}$. We then have

$$\text{len}(\gamma) = \sum_{j=1}^{j^*} \sum_{i \in I_j} \text{len}(\gamma_{i+1}^{(j)}) \leq \sum_{j=1}^{j^*} \sum_{i \in I_j} 2(6 \cdot 2^j)^d \cdot \frac{\lambda(\mathbf{o})}{2^i}.$$

Using that for all $j < j^*$ the sets I_j and I_{j+1} have only $i_j = \max(I_j)$ in common, that $2^j \leq c^*$ for all $j < j^*$, and that $2^{j^*} < 2c^*$, we get

$$\begin{aligned} \text{len}(\gamma) &\leq \sum_{j=1}^{j^*-1} 2(6 \cdot 2^j)^d \cdot \frac{\lambda(\mathbf{o})}{2^{i_j}} + \sum_{i=0}^{\infty} 2(6 \cdot 2c^*)^d \cdot \frac{\lambda(\mathbf{o})}{2^i} \\ &\leq 2 \cdot 6^d \cdot \lambda(\mathbf{o}) \cdot \sum_{j=1}^{j^*-1} (2^d)^j + 4(12 \cdot c^*)^d \cdot \lambda(\mathbf{o}) \\ &\leq 2 \cdot 6^d \cdot \lambda(\mathbf{o}) \cdot \frac{(2^d)^{j^*}}{2^d - 1} + 4(12 \cdot c^*)^d \cdot \lambda(\mathbf{o}) \\ &\leq 4 \cdot 3^d \cdot \lambda(\mathbf{o}) \cdot (2c^*)^d + 4(12 \cdot c^*)^d \cdot \lambda(\mathbf{o}) \\ &\leq 8 \cdot 12^d \cdot (c^*)^d \cdot \lambda(\mathbf{o}) \leq 12^{d+1} \cdot (c^*)^{d+1} \cdot |\mathbf{ot}|. \end{aligned}$$

The path makes an infinite number of straight-line moves. Since for each $i \in \mathbb{N}$, the suffix of the path γ after $p_{i+1}^{(j)}$, for j such that $i+1 \in I_j$, is at distance at most $2 \cdot \lambda(p_{i+1}^{(j)}) \leq 2 \cdot \lambda(\mathbf{o})/2^{i+1} = \lambda(\mathbf{o})/2^i$ from $\mathbf{t}$, Lemma 1 implies that $\mathbf{t}$ is the endpoint of the path γ .

3.1 Extensions

First, the same approach works for spaces of bounded doubling dimension as long as there is a concept of path to connect points whose length is the same as the distance or we change the setting so that the cost of moving from one point to another is the distance between the points. For this, one has to use ε -nets in spaces of bounded doubling dimension [18].

When the target is known to have integral coordinates, we can modify the search so that it has a finite number of segments. Indeed, as soon as we reach a point p_i such that $\lambda(p_i) < 1/2$, we know that the target is at distance smaller than $1/2$ from p_i and there is a unique point with integral coordinates in $B(p_i, \lambda(p_i))$. We can then just move that point. A similar approach works when the target is known to have coordinates with bounded resolution by scaling the setting. In the case where we can detect the target when we are within a given distance δ , we can finish the search when we reach a point p_i with $\lambda(p_i) \leq \delta$, thus, also bounding the number of steps (which will depend on the initial estimate $\lambda(p_0)$).

Our strategies work also in the case where predictions may additionally underestimate the actual distance to the target by, say, some factor $c' \leq 1$, i.e., $c' \cdot |p\mathbf{t}| \leq \lambda(p)$. Then, when both factors c, c' are known, this is equivalent to scaling up the prediction by $1/c'$ to get a new one with factor c/c' . When the factors are unknown, the strategy in Theorem 2 still visits points with geometrically decreasing predictions and since for each point, say p_i , we now have that $|p_i\mathbf{t}| \leq \lambda(p_0)/(2^i \cdot c')$ the path converges to the target (as c' is constant).

One can consider the version when the target is an unknown k -flat F in $\mathbb{R}^d$. The same strategies work, also if we do not know the prediction factor nor the dimension, k , of the flat. Indeed, we are constructing a path as a concatenation of segments such that the distance to F gets arbitrarily small for each suffix of the path. Formally, for each $\varepsilon > 0$, there is a suffix of the path such that all the points on the path are at distance ε from F . Since the path is continuous and has bounded length, the endpoint of the path has to be at distance 0 from F .

Finally, note that for small d there are tighter bounds on the size of ε -nets translating to better constants in our approach. For $c \approx 1$ one can also exploit that we need an ε -net of the spherical shell $S(\mathbf{o}, r/c, r)$, which is much smaller than the whole ball $S(\mathbf{o}, r)$. However, this improvement is expected to be small since even the ε -nets for spheres are not much better.

4 Lower bound

In this section we provide a lower bound for the search problem with c -predictions. Our lower bounds are meaningful when we assume that c is large enough. The

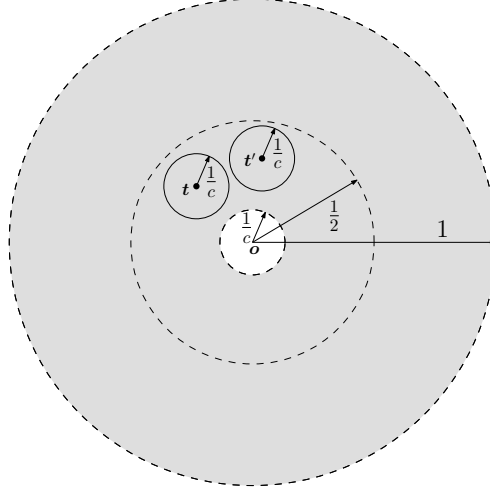


Fig. 4. Parts in the domains for λ_t and $\lambda_{t'}$ for two targets when $d = 2$.

idea is to construct many c -predictions for several different targets that are indistinguishable, unless we are quite close to the target. For each such prediction λ , there is a small ball around the target where the value of λ is different from the other predictions, but all the other predictions have the same value on that small ball. Then, the searcher has to visit *all* the small balls around the targets, because in the worst case the target is going to be in the last small ball visited.

First, using the techniques by Elbassioni, Fishkin and Sitters [12] developed for the Euclidean Traveling Salesperson with Neighbourhoods, one can show the following bound. See Dumitrescu and Tóth [11] for an improvement over [12] where the same idea is reused.

Lemma 4. *Let $\mathbb{B}$ be a set of n congruent balls of radius δ in $\mathbb{R}^d$ that are pairwise interior disjoint. Each path in $\mathbb{R}^d$ that contains at least one point from each ball of $\mathbb{B}$ has length at least $(n/2^d - 1) \cdot \delta \cdot \sqrt{\frac{\pi}{d}}$.*

Next, fix a value $c > 2$. For each $\mathbf{t} \in B(\mathbf{o}, 1/2 - 1/c)$, let $\lambda_{\mathbf{t}}$ be the function defined as follows; see Figures 4 and 5 for intuition.

$$\lambda_{\mathbf{t}}(p) = \begin{cases} c \cdot |p\mathbf{t}|, & \text{if } x \in B(\mathbf{t}, 1/c), \\ 1, & \text{if } x \in B(\mathbf{o}, 1/2) \setminus B(\mathbf{t}, 1/c), \\ 2 \cdot |p\mathbf{o}|, & \text{if } x \notin B(\mathbf{o}, 1/2). \end{cases} \quad (2)$$

Lemma 5. *Assume that $c > 2$ and $|\mathbf{o}\mathbf{t}| \leq \frac{1}{2} - \frac{1}{c}$. Then the function $\lambda_{\mathbf{t}}$ is a c -prediction for the target $\mathbf{t}$. Moreover, for any two distinct $\mathbf{t}, \mathbf{t}'$ satisfying the hypothesis, the functions $\lambda_{\mathbf{t}}$ and $\lambda_{\mathbf{t}'}$ agree on all points outside $B(\mathbf{t}, 1/c) \cup B(\mathbf{t}', 1/c)$.*

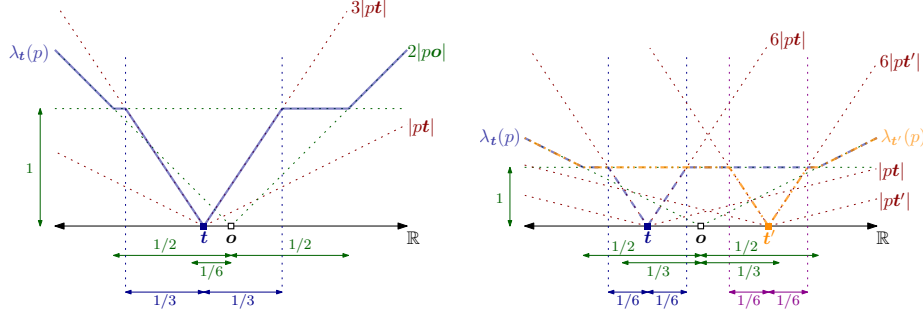


Fig. 5. Examples of functions λ_t for $d = 1$. Note that the axes have different scales. Left: example for $c = 3$; the target has to be at distance at most $1/6$ from the origin. Right: two functions for $c = 6$; the target has to be at distance at most $1/3$ from the origin.

Theorem 3. Assume that $c \geq 4$. There is family of c -predictions for targets in $B(o, 1/4)$ such that any search path in $\mathbb{R}^d$ that uses c -predictions to find the target has length at least $(c^{d-1}/16^d) \cdot \min\{\sqrt{\pi/d}, 1\}$.

Proof. Set $\varepsilon = 2/c$. Because of Lemma 2, the ball $B(o, 1/4)$ has a ε -separated set N with

$$|N| \geq \left(\frac{1/4}{2/c}\right)^d = (c/8)^d$$

points. Note that the balls $B(q, 1/c)$, where $q \in N$, are pairwise interior disjoint.

For each point $t \in N$, consider the function λ_t defined in equation (2). Note that $\frac{1}{4} \leq \frac{1}{2} - \frac{1}{c}$ because $c \geq 4$, and therefore $B(o, 1/4) \subseteq B(o, 1/2 - 1/c)$, as required for the definition. Lemma 5 shows that the function λ_t is a c -prediction for each $t \in N$. Consider the set of prediction functions $A = \{\lambda_t \mid t \in N\}$. Let $\mathbb{B} = \{B(t, 1/c) \mid t \in N\}$ and note that these balls are pairwise interior disjoint. All the functions $\lambda \in A$ take the same value on each point p outside $\bigcup \mathbb{B}$. Moreover, to distinguish between two functions $\lambda_t, \lambda_{t'} \in A$ we have to evaluate them at some point of $B(t, 1/c) \cup B(t', 1/c)$.

Consider any search strategy when the c -prediction function is selected from A by an *adversary*. It may use that the c -prediction is from A . In particular, we know beforehand that the target t is a point of N , selected by the adversary. We claim that the search path has to visit the interior of all the balls of $\mathbb{B}$. Indeed, while there are two distinct points $t, t' \in N$ such that the balls $B(t, 1/c)$ and $B(t', 1/c)$ are not visited by the search path, we have $\lambda_t(p) = \lambda_{t'}(p)$ for all points p along the path, and therefore the information collected cannot discern whether the target is t or t' . Thus, in the worst case, the search path has to visit all the balls of $\mathbb{B}$ but one, to identify which point of N is the target, and then still may have to move to that last ball. That is, we may assume that in the worst case the adversary chooses $\lambda_t \in A$ such that $B(t, 1/c)$ is the *last* ball of $\mathbb{B}$ visited by

the search strategy. After the searcher deduces at which point of N the target is, they still have to move there.

We conclude that the search path has to visit all $|N| \geq (c/8)^d$ balls of $\mathbb{B}$, and for the last ball it still has to travel $1/c$ to the center. Since those balls are pairwise interior disjoint, Lemma 4 implies that the search path has length at least

$$\begin{aligned} \frac{1}{c} + \left(\frac{|N|}{2^d} - 1 \right) \cdot \left(\frac{1}{c} \right) \cdot \sqrt{\frac{\pi}{d}} &\geq \frac{1}{c} + \left(\frac{|N|}{2^d} - 1 \right) \cdot \left(\frac{1}{c} \right) \min\{\sqrt{\pi/d}, 1\} \\ &\geq \frac{|N|}{2^d} \cdot \left(\frac{1}{c} \right) \cdot \min\{\sqrt{\pi/d}, 1\} \\ &\geq \frac{(c/8)^d}{2^d} \cdot \left(\frac{1}{c} \right) \cdot \min\{\sqrt{\pi/d}, 1\} \\ &= \frac{c^{d-1}}{16^d} \cdot \min\{\sqrt{\pi/d}, 1\}. \end{aligned}$$

For the *competitive ratio* we have to compare the length of the search path to the distance to the target, which in the construction is at most $1/4$. We then obtain the following bound, which is interesting when $d \geq 2$ and c is large enough.

Corollary 1. *Consider the search with predictions problem in $\mathbb{R}^d$ with prediction factor $c \geq 4$. Any search strategy to reach the target has competitive ratio at least $(1/4) \cdot (c/16)^{d-1} \cdot \min\{\sqrt{\pi/d}, 1\}$.*

The lower bound still holds in the case where the target has integral coordinates or is detected when we are within a given distance, with a slightly worse constant in the basis of the exponential function. For this, we can just scale the input such that the balls in the construction are large enough or the epsilon-net can be assumed to have points with integer coordinates. Then, we still have to visit all balls in order to detect which one has the target.

A similar lower bound holds for randomized search strategies. In our construction, the searcher has to visit a certain number of disjoint regions (balls) and they can deduce something only when visiting the ball that contains the target. In the worst-case (or adversary) model, the target is going to be in the last ball that is visited. In a randomized setting, all balls would be equally likely to contain the target, and thus the searcher has to visit in expectation half of the balls because visiting the wrong ball does not provide any information. Thus, we have a TSP where you have to visit (in expectation) half of the pairwise disjoint balls, and a similar lower bound applies because the bound is linear in the number of balls.

Acknowledgments. The authors would like to thank Kostas Tsakalidis for suggesting the area of algorithmic problems with predictions and Konrad Swanepoel for providing a simple argument on why a 2-dimensional disk cannot be covered by a curve of bounded length.

This research was funded in part by the Slovenian Research and Innovation Agency (P1-0297, J1-2452, N1-0218, N1-0285), in part by the Erdős Center, and in part by the European union (ERC grant KARST, 101071836; ERC grant 882971, GeoScape). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Bibliography

- [1] Alpern, S., Gal, S.: The theory of search games and rendezvous, International series in operations research and management science, vol. 55. Kluwer (2003). <https://doi.org/10.1007/b100809>
- [2] Angelopoulos, S.: Online search with a hint. In: Lee, J.R. (ed.) 12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference. LIPIcs, vol. 185, pp. 51:1–51:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). <https://doi.org/10.4230/LIPICS.ITCS.2021.51>
- [3] Angelopoulos, S.: Competitive search in the line and the star with predictions. In: Leroux, J., Lombardy, S., Peleg, D. (eds.) 48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, August 28 to September 1, 2023, Bordeaux, France. LIPIcs, vol. 272, pp. 12:1–12:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2023). <https://doi.org/10.4230/LIPICS.MFCS.2023.12>
- [4] Baeza-Yates, R.A., Culberson, J.C., Rawlins, G.J.E.: Searching in the plane. *Inf. Comput.* **106**(2), 234–252 (1993). <https://doi.org/10.1006/INCO.1993.1054>
- [5] Banerjee, S., Cohen-Addad, V., Gupta, A., Li, Z.: Graph searching with predictions. In: Kalai, Y.T. (ed.) 14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA. LIPIcs, vol. 251, pp. 12:1–12:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2023). <https://doi.org/10.4230/LIPICS.ITCS.2023.12>
- [6] Beck, A.: On the linear search problem. *Israel Journal of Mathematics* **2**, 221–228 (1964). <https://doi.org/10.1007/BF02759737>
- [7] Beck, A., D.J., N.: Yet more on the linear search problem. *Israel Journal of Mathematics* **8**, 419–429 (1970). <https://doi.org/10.1007/BF02798690>
- [8] Bellman, R.: Minimization problem. *Bull. Amer. Math. Soc.* **62**, 270 (1956)
- [9] Bellman, R.: Problem 63-9, an optimal search. *SIAM review* **5**(3), 274–274 (1963)
- [10] Bose, P., Carufel, J.D., Durocher, S.: Searching on a line: A complete characterization of the optimal solution. *Theor. Comput. Sci.* **569**, 24–42 (2015). <https://doi.org/10.1016/J.TCS.2014.12.007>
- [11] Dumitrescu, A., Tóth, C.D.: The traveling salesman problem for lines, balls, and planes. *ACM Trans. Algorithms* **12**(3), 43:1–43:29 (2016). <https://doi.org/10.1145/2850418>

- [12] Elbassioni, K.M., Fishkin, A.V., Sitters, R.: Approximation algorithms for the Euclidean traveling salesman problem with discrete and continuous neighborhoods. *Int. J. Comput. Geom. Appl.* **19**(2), 173–193 (2009). <https://doi.org/10.1142/S0218195909002897>
- [13] Few, L.: The shortest path and the shortest road through n points. *Mathematika* **2**(2), 141–144 (1955). <https://doi.org/10.1112/S0025579300000784>
- [14] Gal, S.: A general search game. *Israel Journal of Mathematics* **12**, 32–45 (1972)
- [15] Gal, S.: Minimax solutions for linear search problems. *SIAM Journal on Applied Mathematics* **27**(1), 17–30 (1974)
- [16] Gal, S.: *Search Games*. Academic Press, New York (1980)
- [17] Ghosh, S.K., Klein, R.: Online algorithms for searching and exploration in the plane. *Comput. Sci. Rev.* **4**(4), 189–201 (2010). <https://doi.org/10.1016/J.COSREV.2010.05.001>
- [18] Har-Peled, S., Mendel, M.: Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.* **35**(5), 1148–1184 (2006). <https://doi.org/10.1137/S0097539704446281>
- [19] Hipke, C.A., Icking, C., Klein, R., Langetepe, E.: How to find a point on a line within a fixed distance. *Discret. Appl. Math.* **93**(1), 67–73 (1999). [https://doi.org/10.1016/S0166-218X\(99\)00009-8](https://doi.org/10.1016/S0166-218X(99)00009-8)
- [20] Isbell, J.R.: An optimal search pattern. *Naval Research Logistics Quarterly* **4**, 357–359 (1957)
- [21] Klein, R.: *Algorithmische Geometrie*. Springer-Verlag, Heidelberg, second edn. (2005). <https://doi.org/10.1007/3-540-27619-X>
- [22] Kupavskii, A., Welzl, E.: Lower bounds for searching robots, some faulty. *Distributed Comput.* **34**(4), 229–237 (2021). <https://doi.org/10.1007/S00446-019-00358-Y>
- [23] Langetepe, E.: On the optimality of spiral search. In: Charikar, M. (ed.) *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17–19, 2010*. pp. 1–12. SIAM (2010). <https://doi.org/10.1137/1.9781611973075.1>
- [24] López-Ortiz, A., Schuierer, S.: The ultimate strategy to search on m rays? *Theor. Comput. Sci.* **261**(2), 267–295 (2001). [https://doi.org/10.1016/S0304-3975\(00\)00144-4](https://doi.org/10.1016/S0304-3975(00)00144-4)
- [25] Mitzenmacher, M., Vassilvitskii, S.: Algorithms with predictions. In: Roughgarden, T. (ed.) *Beyond the Worst-Case Analysis of Algorithms*, pp. 646–662. Cambridge University Press (2020). <https://doi.org/10.1017/9781108637435.037>
- [26] Schuierer, S.: On-line searching in simple polygons. In: Christensen, H.I., Bunke, H., Noltemeier, H. (eds.) *Sensor Based Intelligent Robots, International Workshop, Dagstuhl Castle, Germany, September 28 - October 2, 1998, Selected Papers. Lecture Notes in Computer Science*, vol. 1724, pp. 220–239. Springer (1998). https://doi.org/10.1007/10705474_12
- [27] Vaughan, R.C., Wooley, T.D.: The asymptotic formula in Waring’s problem: Higher order expansions. *Journal für die reine und angewandte Mathematik (Crelles Journal)* **2018**(742), 17–46 (2018). <https://doi.org/10.1515/crelle-2015-0098>