Competitive Searching for a Line on a Line Arrangement

Quirijn Bouts
ASML Veldhoven, the Netherlands

Thom Castermans[1]
TU Eindhoven, the Netherlands
t.h.a.castermans@tue.nl

Arthur van Goethem
TU Eindhoven, the Netherlands
a.i.v.goethem@tue.nl

Marc van Kreveld[2]
Utrecht University, the Netherlands
m.j.vankreveld@uu.nl

Wouter Meulemans[3]
TU Eindhoven, the Netherlands
w.meulemans@tue.nl

Abstract

We discuss the problem of searching for an unknown line on a known or unknown line arrangement by a searcher S, and show that a search strategy exists that finds the line competitively, that is, with detour factor at most a constant when compared to the situation where S has all knowledge. In the case where S knows all lines but not which one is sought, the strategy is 79-competitive. We also show that it may be necessary to travel on $\Omega(n)$ lines to realize a constant competitive ratio. In the case where initially, S does not know any line, but learns about the ones it encounters during the search, we give a 414.2-competitive search strategy.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Competitive searching, line arrangement, detour factor, search strategy

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2018.49

Acknowledgements This work was started during the 2nd AGA Workshop, in Jan./Feb. 2017. The authors thank two anonymous reviewers for their extensive and detailed comments.

1 Introduction

Given a set L of n lines $\ell_0, \ell_1, \dots \ell_{n-1}$ in the plane, consider the arrangement $\mathcal{A}$ that they form as a geometric graph. Technically, $\mathcal{A}$ is not a graph due to half-infinite edges, but in our problem we can end each line at its extreme intersection points, and hence we can use the term graph without complications. We consider paths on $\mathcal{A}$. The cost of a path on $\mathcal{A}$ is the Euclidean length of that path. The distance between two points on $\mathcal{A}$ is the cost (or length) of the shortest path that stays on $\mathcal{A}$ between those points.

[1] The Netherlands Organisation for Scientific Research (NWO) supports T.C. under project no. 314.99.117.
[2] Supported by the Netherlands Organisation for Scientific Research on grant no. 612.001.651.
[3] Supported by the Netherlands eScience Center (NLeSC) on project 027.015.G02.

29th International Symposium on Algorithms and Computation (ISAAC 2018).
Editors: Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao; Article No. 49; pp. 49:1–49:12
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Assume that a searcher S is located on some vertex or edge of the graph. Denote its initial position by O. The searcher S can only travel on the arrangement and is hence restricted to paths on $\mathcal{A}$. Searcher S is looking for a target line $\ell_t \in L$, but does not know which of the lines in L corresponds with ℓ_t. The searcher S will recognize ℓ_t when it reaches any point on ℓ_t (necessarily at an intersection point with another line). We call this special line the *target line*, and assume that O does not lie on ℓ_t. If it would, the problem would be solved immediately. We consider two versions of the problem: one where S knows the lines in L and therefore $\mathcal{A}$ completely, and one where S only knows about the existence and parameters of a line once it reaches some point on it.

We will show that a search strategy exists by which S can reach the target line *competitively* in both versions. In other words, S can reach the target line with a detour factor bounded by a constant, when compared to the shortest path on $\mathcal{A}$ to the target line. Competitive analysis is commonly used to compare "the cost of not knowing" with "the cost of knowing". The maximum detour factor of a search strategy is known as its *competitive ratio*. The *competitive ratio of a search problem* is the infimum of the competitive ratios of all search strategies that solve that search problem.

The best known search problem is perhaps the one-dimensional problem of finding a point on a line from a starting position. If we know the distance d, but not whether it is to the left or to the right, the optimal strategy is to go left over a distance d and then right over a distance $2d$. We find the point with competitive ratio 3, which is optimal. If we don't know the distance but we do know some (very) small lower bound ϵ on the distance, it is best to go ϵ to the left, then back and another 2ϵ to the right, then back and another 4ϵ to the left, and so on. This doubling strategy gives a competitive ratio of 9, which is known to be optimal as proved by Beck and Newman [4] in 1970, see also [2, 13].

The problem of searching for a line in the plane without obstacles was studied by Baeza-Yates et al. [2] in various settings. The settings refer to the knowledge we have of the line, which can be its slope, its distance, both, or neither. If the slope of the line is known, the problem reduces to the one-dimensional problem just discussed. If only the distance is known, the optimal competitive ratio is 6.39.... The problem of searching for a line a given distance away was posed by Bellman [5] in 1956 and solved by Isbell [20] in 1957. It is a classic in recreational mathematics and often posed as a swimmer in the fog, trying to reach the (straight) shore which is a known unit distance away, while swimming the least in the worst case. If the slope nor the distance of a line to be found is known, the best known competitive ratio is 13.81..., which is realized by a logarithmic spiral search strategy.

Competitive analysis of algorithms was introduced by Sleator and Tarjan for analyzing the list update problem [24]. Here the lack of knowledge is the next online requests. In geometric situations, the lack of knowledge is often the environment itself or the location of something to be found (by seeing or reaching it). The main motivation of such problems comes from the navigation of robots in unknown environments. More generally, searching for a target in environments where either the target or the environment is unknown is a basic problem, and competitive analysis is a fundamental way to understand what is in principle possible in such exploration problems. We list a few main results on searching and competitive analysis in geometric and geometric-graph environments; for an extensive overview see also [15]. We begin by noting that there is no c-competitive search strategy to find an unknown target node in a known graph, for example when the graph is a star.

When searching for an unknown target on a line, but additional information on the distance to the target is known, alternative results can be obtained [8, 17]. Demaine et al. [13] show that searching for an unknown target on a line with cost depending on both

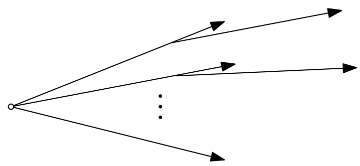

Figure 1 Half-lines cannot be searched c-competitively.

search distance and turns can be done competitively with cost $9OPT + 2d$, where d is the cost of one turn. Searching on multiple rays is studied in various papers [8, 13, 16, 23]; Kao et al. [22] give an optimal randomized algorithm. In yet other variants one can search with multiple searchers [3, 16].

Kalyanasundaram and Pruhs [21] consider visibility-based searching for a recognizable point in an unknown scene with convex obstacles. Their result on the competitive factor is not constant, but depends on the number of obstacles and their aspect ratio. Blum et al. [6] investigate similar problems for different classes of obstacles. Hoffmann et al. [18] show that an unknown simple polygon can be discovered completely with a competitive ratio of 26.5. There are various other visibility-based search problems addressed with competitive analysis (e.g.,[14, 19]).

A different setting where competitive strategies are investigated is routing in geometric graphs. Here an unknown geometric graph is given along with a source and target with known coordinates. We route a package from source to target over the nodes, but learn about the existence and coordinates of a node when we are at a neighbor. For triangulations, no c-competitive strategy exists, but for special triangulations like Delaunay and certain other geometric graphs, a constant competitive strategy does exist [7, 9, 10, 11, 12]. Searching for an unknown target on a planar straight line graph with discovery based on Pókemon Go was investigated with competitive analysis recently [25].

Contributions. In Section 2 we give a preliminary result where we use only two lines and obtain a competitive ratio depending on their angle. Moreover, we show that, if we want to obtain a *constant* competitive ratio that does not depend on parameters of the arrangement, then the search strategy must allow for traversing at least half the lines in an arrangement. In Section 3 we describe and analyze such a strategy and show that this leads to a 79-competitive strategy. This is an upper bound on the relative cost of not knowing which line is sought. (Note that for slightly more complex objects like half-lines, no constant-competitive strategy exists by mimicking a star graph, see Figure 1.) In Section 4 we generalize the problem to the situation where the searcher does not know all lines beforehand. They learn about the existence of a line and its parameters only when the line is reached. We show that in this case a search strategy exists with competitive ratio 414.2. This is an upper bound on the relative cost of not knowing the lines at all.

Although our search problems and competitive ratios are new, the existing literature implies lower bounds for our versions. When all lines are known, we have a lower bound of 9, because the problem is at least as hard as the one-dimensional problem of finding a point on a line. Moreover, it is essentially also at least as hard as finding a fully unknown line in the plane, because we could be given a very dense set of lines where all movement is approximately possible and every line could be the target. The best known competitive ratio is 13.81... to find an unknown line, but this is not known to be optimal so it does not provide a true lower bound. In case we do not know the lines of the arrangement at all, we inherit the lower bound of searching on four rays (half-lines) for a point, which is 19.96... [1, 13]. The

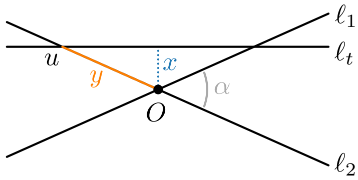

Figure 2 Sketch of worst case.

line arrangement consists of two perpendicular lines, we start on their intersection, and we must explore. If we do not follow the optimal strategy for four rays, the target line was just out of reach at the place where we went less far, and perpendicular to that ray. With more than four rays, lines will intersect more than one ray and the argument no longer works.

2 Competitive searching on an arrangement

As a warm-up, assume that S starts at the intersection of two lines ℓ_1 and ℓ_2 whose smaller intersection angle is $\alpha \leq \pi/2$. Furthermore, S only traverses ℓ_1 and ℓ_2, disregarding all other lines for traversal.

▶ **Theorem 1.** *The target line can be found with competitive ratio at most* $^{29}/_{\sin(\alpha/2)}$.

Proof. Denote the starting point by O and the target line by ℓ_t. As a lower bound for reaching ℓ_t we use the Euclidean distance between O and ℓ_t, denoted by x, because a line ℓ_3 through O and normal to ℓ_t could exist.

Note that ℓ_t must intersect at least one of ℓ_1 and ℓ_2. Let y be the distance on ℓ_1 or ℓ_2 to the closest intersection point u of ℓ_t with ℓ_1 and/or ℓ_2. Since α is the smaller angle, the worst case occurs when the target line ℓ_t spans a triangle with the two initial lines ℓ_1 and ℓ_2 with an angle of $\pi - \alpha$; the worst ratio between x and y occurs when this triangle is equilateral with apex O. This is illustrated in Figure 2. By elementary geometry, we then have $y \leq {^x}/_{\sin(\alpha/2)}$.

The strategy to find ℓ_t is as follows. Let d be the distance between O and the vertex v on ℓ_1 or ℓ_2 closest to it. First, S travels to v and back to O. Then S travels the same distance d in each of the other three directions on ℓ_1 and ℓ_2, and back to O each time. After that we double d and repeat. S has achieved its goal when it reaches u, and therefore ℓ_t.

We can view the traversal of S on ℓ_1 and ℓ_2 as the traversal on four half-lines induced by O. One of these half-lines crosses ℓ_t at distance y. This is, by definition, where u is. By the doubling strategy, S will have traversed a total distance less than $5y$ on the half-line with u. On each of the other half-lines, S has traversed at most a distance of $8y$. Summing up yields that the searcher travelled at most a distance of $29y$; using $y \leq {^x}/_{\sin(\alpha/2)}$, we find that the competitive ratio, bounded by $^{29y}/_x$, gives the claimed bound of $^{29}/_{\sin(\alpha/2)}$. ◀

We note that a tighter analysis of the same strategy will give a slightly better competitive ratio, and a different strategy where we traverse the half-lines over different distances will also give a better competitive ratio. However the strategy is not c-competitive for any constant c, since α can be arbitrarily small. Moreover, since this is a special case of the problem, we explore this strategy no further.

Below, we show that for any constant c, any c-competitive strategy must traverse $\Omega(n)$ lines. So the strategy of the proof of Theorem 1 cannot work, not even with the usage of some carefully chosen additional lines besides ℓ_1 and ℓ_2.

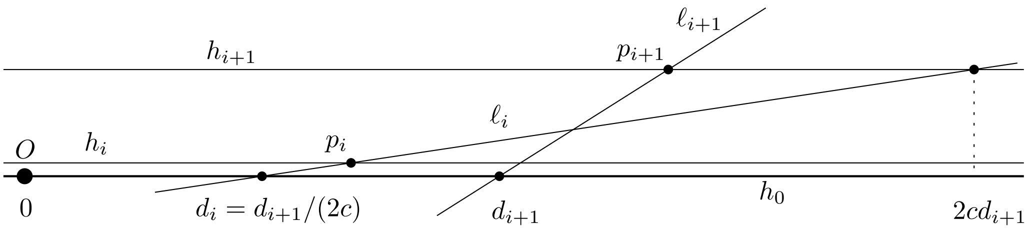

Figure 3 Placement of ℓ_i and h_i, given ℓ_{i+1} and h_{i+1}. Line ℓ_i is defined by the point with x-coordinate $d_{i+1}/(2c)$ on h_0 and the point with x-coordinate $2cd_{i+1}$ on h_{i+1}. Line h_i is placed such that $\mathrm{dist}(h_0 \cap \ell_{i+1}, p_i) < d_{i+1}/(2c)$.

▶ **Theorem 2.** *For any constant $c \geq 1$, there is an arrangement $\mathcal{A}$ of n lines such that any c-competitive strategy must traverse at least $n/2 = \Omega(n)$ lines of $\mathcal{A}$ in the worst case.*

Proof. We construct an arrangement $\mathcal{A}$ of $n = 2m + 1$ lines. The line h_0 is the x-axis, and searcher S starts on h_0 at the origin O. Let $h_1, \dots, h_m$ be horizontal lines that together with h_0 have a bottom-to-top order $h_0, h_1, \dots, h_m$. Let $\ell_1, \dots, \ell_m$ be m lines with positive slope ≤ 1, such that the upper envelope of $\ell_1, \dots, \ell_m$ is a convex increasing function that contains all these lines in the same order. We ensure that these lines intersect h_0 on the positive side and in the order $\ell_1, \dots, \ell_m$. The construction will be such that the part of ℓ_i between its intersection with h_i and its intersection with ℓ_{i-1} must be used by S to reach h_i with detour no more than c, because even the intersection of ℓ_{i-1} with h_i has an x-coordinate that is too high.

In more detail, we construct the lines incrementally from m down to 1, in pairs ℓ_i and then h_i, see Figure 3. We start with $\ell_m : y = x - 2$ and $h_m : y = 1$. Assume ℓ_{i+1} and h_{i+1} are placed, and their intersection point p_{i+1} is such that $d_{i+1} = \mathrm{dist}(O, h_0 \cap \ell_{i+1}) > \mathrm{dist}(h_0 \cap \ell_{i+1}, p_{i+1})$ (for ℓ_m and h_m we made sure of this condition). Then we define ℓ_i by constructing two points on it. One is the point $(d_{i+1}/(2c), 0)$ on h_0; the other is the point on h_{i+1} with x-coordinate $2cd_{i+1}$. This defines ℓ_i. The line h_i is chosen horizontal and low enough so that $\mathrm{dist}(h_0 \cap \ell_i, p_i) < \mathrm{dist}(O, h_0 \cap \ell_i) = d_{i+1}/(2c)$. Note that $d_m = 2$ and $d_i = 2/(2c)^{m-i}$.

To argue that this arrangement forces a searcher S to walk on every line ℓ_i (and also h_0 where S starts), we observe that we can reach the line h_i in distance at most d_{i+1}/c simply by following h_0 and ℓ_i only (we can do a little bit better but for the proof this is not needed). To reach h_i c-competitively we must thus travel less than d_{i+1} along $\mathcal{A}$.

We cannot use line ℓ_{i+1} or any higher-indexed line, because all their vertices have x-coordinates at least d_{i+1} so it must take d_{i+1} or more to even reach ℓ_{i+1} or a later line. Thus if we do not use ℓ_i, we must reach line h_i on line ℓ_{i-1} or a lower-indexed line. By construction the intersection of ℓ_{i-1} and h_i has x-coordinate d_{i+1}. Thus reaching h_i from ℓ_{i-1} is not c-competitive. Furthermore, any line ℓ_j with $1 \leq j < i - 1$ must intersect h_i right of the intersection with ℓ_{i-1} and thus for the same reason reaching h_i via ℓ_j cannot be c-competitive.

In other words, we must use ℓ_i to get c-competitively to h_i, and any of the m horizontal lines $h_1, \dots, h_m$ can be the target line. Hence, a c-competitive strategy must visit and walk on each of $\ell_1, \dots, \ell_m$. As S starts on h_0, it thus walks on at least $m + 1 \geq n/2$ lines. ◀

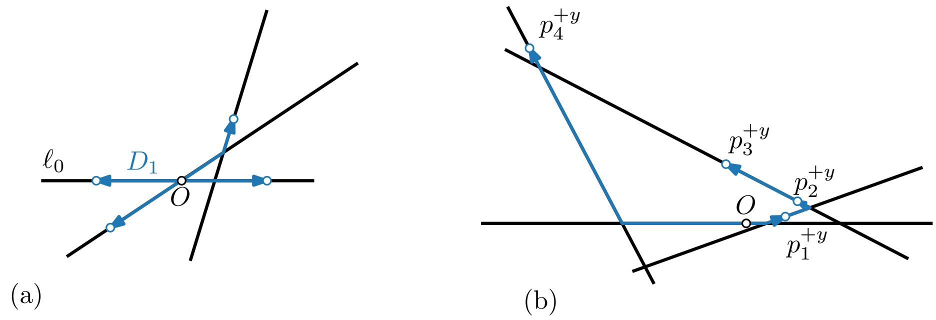

Figure 4 (a) Explored paths of length D_1 reaching the maximum (minimum) x- and y-coordinate. (b) The paths of doubling lengths $D_1, \ldots, D_4$ to the highest points $p_1^{+y}, \ldots, p_4^{+y}$.

3 A c-competitive search strategy on a known arrangement

We continue with the general case where S may start anywhere on any line and we make no assumptions on the angles between intersecting lines. For convenience we will assume the starting point to be at the origin O and the line crossing through O to be ℓ_0. If multiple lines cross the origin, we pick ℓ_0 to be the line that intersects any other line closest to O. We will assume ℓ_0 is horizontal. As the problem is rotation and translation invariant these assumptions do not change the problem. As before let d be the distance to the closest intersection point on ℓ_0.

Consider the following search strategy for S. Searcher S iteratively explores the arrangement starting from the origin. In iteration i four paths of length $2^i \cdot d$ are explored starting at O. These paths are picked such that they maximize (minimize) the x- respectively y-coordinate that S can achieve on the arrangement within distance $2^i \cdot d$ from O (see Figure 4(a)). Specifically this results in the following strategy. First, S traverses ℓ_0 over a distance $2d$ in the direction $+x$ and then returns back to O. Second, S traverses ℓ_0 for a distance $2d$ in the direction $-x$ and back. Third, S determines the point on $\mathcal{A}$ with maximum y-coordinate it can reach when traversing over a distance $2d$; S goes there and back. Symmetrically, S also visits the point with lowest y-coordinate reachable within distance $2d$ from O. Upon returning to the origin the allowed distance is doubled and the process is repeated until S finds ℓ_t.

Let D_i be the distance travelled in iteration i. Let the points where S ends when searching over a distance D_i with minimum and maximum x- and y-coordinate be denoted p_i^{-x}, p_i^{+x}, p_i^{-y}, and p_i^{+y}, respectively. Figure 4(b) shows the four paths to $p_1^{+y}, \ldots, p_4^{+y}$. Notice that the path for D_{i+1} does not necessarily follow the path for D_i as a less steep line may be followed to reach a steeper line sooner.

▶ **Lemma 3.** *The y-coordinate of p_i^{+y} is at least twice that of p_{i-1}^{+y}. The symmetric statement holds for p_i^{-y} and p_{i-1}^{-y}.*

Proof. Observe that for any p_{i-1}^{+y}, the last line traversed on the path to p_{i-1}^{+y} must have the steepest absolute slope. If not, we could get higher by staying on the line with steepest slope. When we traverse a distance D_i instead of D_{i-1}, we have the option of staying on this steepest absolute slope line, and since $D_i = 2D_{i-1}$, we get at least twice as high just by staying on the line that contains p_{i-1}^{+}. ◀

Let Q_i be the convex quadrilateral with the points p_i^{-x}, p_i^{+x}, p_i^{-y}, and p_i^{+y} as vertices and let R_i be the axis-parallel rectangle with these four points on its boundary (see Figure 5).

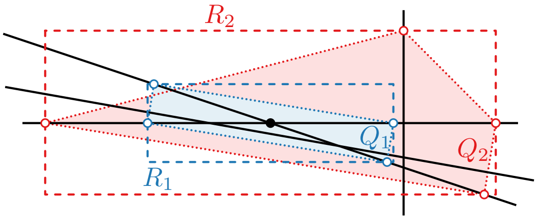

Figure 5 Illustration of the quadrilaterals Q_1 and Q_2 and the respective axis-parallel bounding rectangles R_1 and R_2. Notice that consecutive quadrilaterals need not be contained in each other.

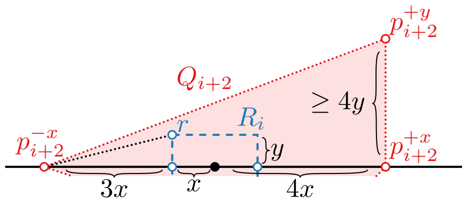

Figure 6 Even with the (impossible) worst-case placement of p_{i+2}^{+y} rectangle R_i is still contained in Q_{i+2}.

Trivially $Q_i \subset R_i$ and from Lemma 3 it immediately follows that $R_1 \subset R_2 \subset \cdots \subset R_k$.

▶ **Lemma 4.** $R_i \subset Q_{i+2}$.

Proof. Without loss of generality only consider the half-plane above ℓ_0. We show that the triangle $p_{i+2}^{-x}p_{i+2}^{+y}p_{i+2}^{+x}$ contains the rectangle with bottom vertices p_i^{-x} and p_i^{+x} and top side through p_i^{+y}. We know that $p_{i+2}^{-x}p_{i+2}^{+x}$ is exactly four times the length of $p_i^{-x}p_i^{+x}$ as ℓ_0 is horizontal. By Lemma 3 the y-coordinate of p_{i+2}^{+y} is at least four times that of p_i^{+y} (see Figure 6). By triangle inequality the x-coordinate of p_{i+2}^{+y} must be between p_{i+2}^{-x} and p_{i+2}^{+x}.

Let x be the x-coordinate of p_i^{+x}, and $r = (-x, y)$ the vertex at the top-left corner of R_i. Consider the side $p_{i+2}^{-x}p_{i+2}^{+y}$ of the triangle and the line $p_{i+2}^{-x}r$. The slope of $p_{i+2}^{-x}r$ is $y/(3x)$. The slope of $p_{i+2}^{-x}p_{i+2}^{+y}$ depends on the exact location of p_{i+2}^{+y}. In the (impossible) worst case p_{i+2}^{+y} is located at $(4x, 4y)$. Thus the slope of $p_{i+2}^{-x}p_{i+2}^{+y}$ is at least $y/(2x)$ and r is below $p_{i+2}^{-x}p_{i+2}^{+y}$. Containment of R_i in Q_{i+2} trivially follows. ◀

We observe that if the target line ℓ_t intersects Q_i then ℓ_t will be found in iteration i or before. Hence the distance travelled by the searcher is upper-bounded by the distance travelled up to and including iteration i. Suppose the searcher S finds the target line ℓ_t in iteration k. We will use the rectangle R_{k-3} as a lower bound on the length of the shortest path to ℓ_t to prove an upper bound on the competitive ratio.

▶ **Lemma 5.** *The target line ℓ_t intersects Q_k and does not intersect R_{k-3}.*

Proof. If ℓ_t intersects Q_{k-1}, then ℓ_t would have been found in phase $k-1$. Since $R_{k-3} \subset Q_{k-1}$, the lemma follows. ◀

As ℓ_t does not intersect R_{k-3} the closest point of ℓ_t to O must be outside of R_{k-3}. But then the shortest path to ℓ_t must have length larger than D_{k-3}. Assume for contradiction that the closest point p_t on ℓ_t has distance less than D_{k-3}. As in iteration $k-3$ we followed the paths that maximize (minimize) the x- and y-coordinate, p_t could be reached and must thus be contained in R_{k-3}. Contradiction. Thus D_{k-3} is a lower bound on the distance from O to ℓ_t, and $D_{k-3} = D_k/8$.

For an upper bound, we consider the distance we have travelled. Except for the last iteration, we traversed four paths of length D_i in two directions in each iteration. Thus in previous iterations we traversed $8\sum_{i=1}^{k-1} D_i$. In the last iteration in the worst-case we discover ℓ_t while traversing the fourth path all the way to its end. Hence we traverse three paths of length D_k twice, and the last path of length D_k once. The total travel is thus at most:

$$8\sum_{i=1}^{k-1} D_i + 7D_k$$

Using the summation $\sum_{i=0}^{k-1} z^i = \frac{z^k-1}{z-1}$ and $D_i = 2^i d$ we can rewrite this to $15 \cdot 2^k d - 16d < 15D_k$. We thus upper-bound the competitive ratio by 120.

A more careful analysis shows that Lemma 4 is true even if we do not double D_i but enlarge by only a factor $\sqrt{3}$. Let $D_1 = \sqrt{3}d$ and $D_i = \sqrt{3}D_{i-1}$ for $i \geq 2$, so $D_i = \sqrt{3}^i \cdot d$, and suppose S finds ℓ_t in iteration k. Then $D_{k-3} = \sqrt{3}^{k-3} d$ is a lower bound for reaching ℓ_t. With the described strategy S travels at most

$$8 \sum_{i=1}^{k-1} \sqrt{3}^i d + 7\sqrt{3}^k d < 8 \frac{\sqrt{3}^k d}{\sqrt{3}-1} + 7\sqrt{3}^k d$$

The competitive ratio becomes

$$\frac{8 \frac{\sqrt{3}^k d}{\sqrt{3}-1} + 7\sqrt{3}^k d}{\sqrt{3}^{k-3} d} = (\frac{8}{\sqrt{3}-1} + 7)\sqrt{3}^3 < 94$$

Another improvement comes from organizing the four traversals in a phase conveniently so that we do not have to go back to O at the end. In every even phase i we start with going to p_i^{+x}, then we do p_i^{+y} and p_i^{-y} in any order, and end with going to p_i^{-x}. In every odd phase j we go to p_j^{-x} first and to p_j^{+x} last. It is easy to see that we do not have to go back at the end of any phase, because we go out over the exact same stretch in the next phase anyway. Instead of traversing $8D_i$ in a phase i, we now traverse $(7 - 1/\sqrt{3}) \cdot D_i$. This also holds for the last phase D_k. With some basic calculation we obtain:

▶ **Theorem 6.** *A* 79*-competitive search strategy exists to find an unknown target line in an arrangement of lines.*

Alternatively, we may also triple D_i because then $R_i \subset Q_{i+1}$; a lower constant factor than 3 will not ensure that $R_i \subset Q_{i+1}$ so that will not give improvements. The competitive ratio we get is worse, however, than when using $\sqrt{3}$ and $R_i \subset Q_{i+2}$.

We note that if we know the exact distance to the line, we can use some of the ideas just given. By the observations above, we can find the unknown line by going three times as far in each direction. For the last direction S does not need to go back, so in total we will find the line with competitive ratio 21.

4 A c-competitive search strategy on an unknown arrangement

In this section we consider the situation where the searcher S does not know the arrangement beforehand. In particular, we assume S learns the slope and intercept of a line, only when S reaches it. The question arises whether we can adapt our competitive strategy to still realize a constant competitive ratio. The exact same strategy cannot be used, because we can no longer determine the points p^{+y} and p^{-y} before we start walking.

First of all, this problem suffers from a technicality that has been observed in similar problems: as soon as we decide to walk *any* distance from the starting location in some direction on the starting line, the target line could have been arbitrarily much closer in the other direction [2]. So a constant competitive ratio cannot exist. This technicality is commonly circumvented by assuming that the target line is at least some known – possibly extremely small – distance away from the start. We will assume this as well.

Assume the starting location is at the origin O and lies on a horizontal line ℓ_0. We start by finding the closest intersection to O. If it is at distance d, then we let $D_1 = 2d$. Similar to the strategy for known arrangements in iteration i we aim to find the leftmost, rightmost,

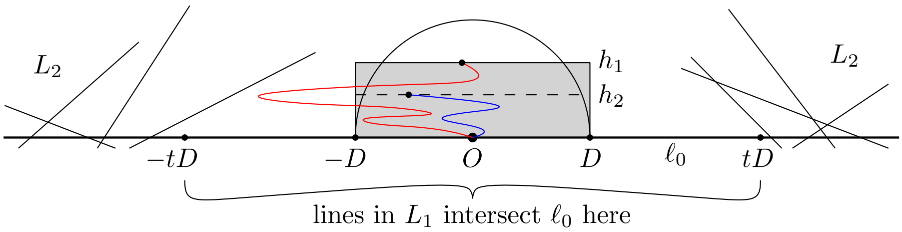

Figure 7 The line sets L_1 and L_2, only some lines in L_2 are shown. Two paths maximizing the achieved height in the vertical slab $[-D, D]$: A path on $L_1 \cup L_2$ of length D (blue) reaching height h_2 and a path on L_1 of length $2tD + 2D$ (red) reaching height h_1. We show $h_1 \geq h_2$.

lowest, and highest point we can reach with distance D_i. We, however, choose our movement as to also discover a suitable set of "nearby" lines to which we must necessarily restrict our movement as we do not know about the existence of other lines. We show that with this smaller set of lines we can still achieve the height that we could have reached with knowledge of all lines; however, we traverse a constant factor further to ensure this.

We start by walking left and right from O over a distance tD for some constant $t \geq 1$ to be specified later. In doing so, we discover a subset L_1 of the lines. Let $L_2 = L \setminus L_1$, see Figure 7. Let h_2 be the height we could achieve within distance D if we had full knowledge of the arrangement. Let the sequence of lines used to reach h_2 be $\ell_0, \ell_1, \ell_2, \ldots, \ell_j$. We know that ℓ_j is the steepest line among these, by the proof of Lemma 3.

We want to reach the highest point in the vertical slab $[-D, D]$ using lines from L_1 only. Clearly within a distance D we can get at most as high as h_2. Instead we allow a traversal of distance $2tD + 2D$ along the lines of L_1. Let h_1 be the maximum height we can achieve while ending in the vertical slab $[-D, D]$ and when travelling over distance at most $2tD + 2D$ along only lines of L_1.

▶ **Lemma 7.** *$h_2 \leq h_1$ if $t \geq 2$.*

Proof. Assume for contradiction that $h_2 > h_1$. Let $\ell_0, \ell_1, \ldots \ell_j$ be the lines on a path of length D to height h_2 on $L = L_1 \cup L_2$. Either $\ell_j \in L_1$ or $\ell_j \in L_2$.

Assume first that $\ell_j \in L_1$. Specifically then there is a point p we can reach along ℓ_j that lies in the slab $[-D, D]$ at height h_2. However, ℓ_j intersects ℓ_0 at most tD from the origin. Thus we can follow ℓ_0 to the intersection with ℓ_j, and then follow ℓ_j to height h_2. As $h_2 \leq D$ this takes at most $tD + (t + 1)D$ horizontal movement and D vertical movement (see Figure 8). The total distance traversed along lines from L_1 is upper bounded by $2tD + 2D$, therefore $h_1 \geq h_2$. Contradiction.

Next, assume that $\ell_j \in L_2$. The line ℓ_j must intersect the rectangle $[-D, D] \times [0, h_2]$ since the path of length D reaching h_2 cannot leave this rectangle. The maximum slope of a line $\ell_j \in L_2$ that intersects this rectangle is $\frac{h_2}{(t-1)D}$ as such a line must intersect ℓ_0 at least tD from the origin.

We must have that ℓ_j has the steepest absolute slope. If a previously traversed line had a steeper absolute slope we could follow it to get higher while staying in the slab $[-D, D]$. Thus the largest (absolute) slope of any line traversed to get to h_2 is $\frac{h_2}{(t-1)D}$. Take $t \geq 2$, then the largest slope is at most $\frac{h_2}{D}$. In the (unachievable) best case we traverse this slope for the full length of the path to height h_2, however then we still reach a height less than h_2. Contradiction. ◀

Our constant competitive strategy, using $t = 2$, is therefore as follows: Go left over $2D$, then right over $4D$, then back to the starting point over $2D$, and form the set L_1.

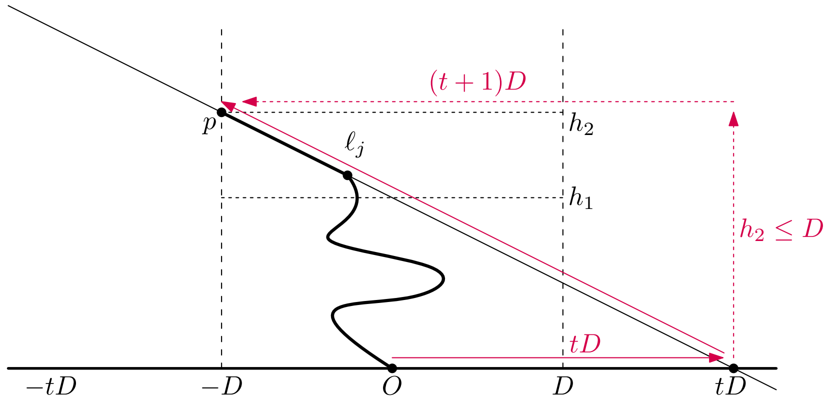

Figure 8 Assume for contradiction that $h_2 > h_1$. The last line traversed to get to height h_2 within distance D on $L_1 \cup L_2$ must then be from L_2. If $\ell_j \in L_1$ then $h_1 \geq h_2$ as we can traverse only ℓ_0 and ℓ_j to reach the same height within distance $2tD + 2D$.

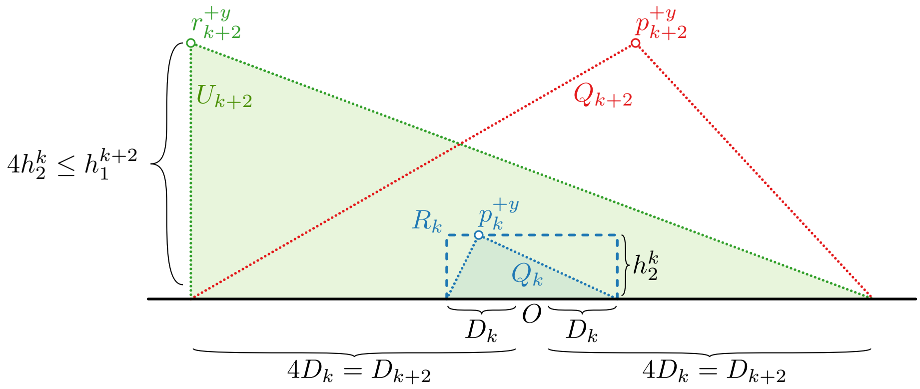

Figure 9 Even with the worst-case placement of r^{+y}_{k+2}, R_k is still contained in U_{k+2}.

Use these lines, using distance $6D$ to get as high as possible in the vertical slab $[-D, D]$, and the same distance to get as low as possible, and back. In total we traverse a distance $8D + 12D + 12D = 32D$ in one phase. Then double D and repeat.

We once again argue that the true minimum and maximum x and y coordinates reachable in some phase i are covered completely by a quadrilateral on the discovered minima and maxima in a later phase. Let U_k be the quadrilateral created by our exploration of four paths on L_1 in phase k.

▶ **Lemma 8.** $R_k \subset U_{k+2}$

Proof. The proof of the lemma is identical to the proof of Lemma 4, with the following minor changes. See Figure 9 for an illustration of the proof.

Let r^{+y}_i be the highest point reachable in the slab $[-D_i, D_i]$ during phase i. Once again let p^{+y}_i be the highest point achievable in distance D_i on the complete arrangement. From Lemma 7 we conclude that the y-coordinate of p^{+y}_{k+2} is less or equal than that of r^{+y}_{k+2}. We also know that the x-coordinate of r^{+y}_{k+2} lies in the slab $[-D_{k+2}, D_{k+2}]$ so we do not need the triangle inequality of the proof. The proof follows directly. ◀

We can now use the same method of analysis as for the case of a fully known line arrangement, except that we have to take into account that the searcher must move more in every phase. Once again we can scale the distance walked in an iteration by a factor of $\sqrt{3}$ instead of 2 to improve the bound. For a line found in iteration i we traverse at most:

$$32 \sum_{i=1}^{k-1} D_i + 36 D_k < 32 \frac{\sqrt{3}^k d}{\sqrt{3} - 1} + 36 \sqrt{3}^k d$$

A line found in iteration i is at least at a distance of $D_{k-3} = \sqrt{3}^{k-3} d$. Thus we obtain the following result.

▶ **Theorem 9.** *A 414.2-competitive search strategy exists to find an unknown target line in an unknown arrangement of lines, where a line becomes known once we reach it.*

5 Conclusions

We have developed and analyzed search strategies for reaching an unknown target line in an arrangements of lines. We did so by considering the competitive ratio: the worst-case ratio between the distance travelled by the searcher and the length of the shortest path from the searcher's start location to the target line. We gave a search strategy for the case of known arrangements that achieves a competitive ratio of 79. Then we generalized our strategy so that it is competitive on line arrangements that are not known beforehand. The parameters of a line become known only when the line is reached. In this case we gave a 414.2-competitive search strategy. There is a considerable gap between the known lower bounds and upper bounds.

Future work. In our work we assumed that the speed on every line is the same. When we drop this assumption we do not know whether searching for a line can be done competitively even if we know all lines and all speeds. Certain properties still hold, for example, if we search for the largest y-coordinate, then we can get twice as far if we double the travel time. However, a diagonal with high speed may cause the furthest reachable point in both horizontal and vertical direction to be along this diagonal, essentially preventing growth of the explored region into other directions. When we search with a cost T from O, the relevant points to visit seem to be the vertices of a convex polygon that is the convex hull of all points reachable at cost T. This polygon can have more than constantly many vertices so we cannot visit all in a phase. It is unclear how to choose a constant-size subset so that the resulting, smaller convex hull at least contains the full convex hull from a previous iteration.

We note that searching (connected) arrangements of simple geometric objects like line segments, circles, and half-lines cannot be done with a constant competitive strategy. But it is possible that if we impose restrictions on the arrangement, constant-competitive search strategies can be developed.

References

1 Steve Alpern and Shmuel Gal. *The Theory of Search Games and Rendezvous*, volume 55. Springer Science & Business Media, 2006.
2 Ricardo Baeza-Yates, Joseph Culberson, and Gregory Rawlins. Searching in the Plane. *Information & Computation*, 106(2):234–252, 1993.
3 Ricardo Baeza-Yates and René Schott. Parallel searching in the plane. *Computational Geometry*, 5(3):143–154, 1995.
4 Anatole Beck and D.J. Newman. Yet more on the linear search problem. *Israel Journal of Mathematics*, 8(4):419–429, 1970.

5 Richard Bellman. A minimization problem. *Bulletin of the American Mathematical Society*, 62(3):270, 1956.

6 Avrim Blum, Prabhakar Raghavan, and Baruch Schieber. Navigating in unfamiliar geometric terrain. *SIAM Journal on Computing*, 26(1):110–137, 1997.

7 Prosenjit Bose, Andrej Brodnik, Svante Carlsson, Erik D. Demaine, Rudolf Fleischer, Alejandro López-Ortiz, Pat Morin, and J. Ian Munro. Online routing in convex subdivisions. *International Journal of Computational Geometry and Applications*, 12(4):283–295, 2002.

8 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.

9 Prosenjit Bose, Jean-Lou De Carufel, Stephane Durocher, and Perouz Taslakian. Competitive online routing on Delaunay triangulations. *International Journal of Computational Geometry & Applications*, 27(04):241–253, 2017.

10 Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. Optimal local routing on Delaunay triangulations defined by empty equilateral triangles. *SIAM Journal on Computing*, 44(6):1626–1649, 2015.

11 Prosenjit Bose and Pat Morin. Competitive online routing in geometric graphs. *Theoretical Computer Science*, 324(2-3):273–288, 2004.

12 Prosenjit Bose and Pat Morin. Online routing in triangulations. *SIAM Journal on Computing*, 33(4):937–951, 2004.

13 Erik D. Demaine, Sándor P. Fekete, and Shmuel Gal. Online searching with turn cost. *Theoretical Computer Science*, 361(2-3):342–355, 2006.

14 Sándor P. Fekete, Rolf Klein, and Andreas Nüchter. Online searching with an autonomous robot. *Computational Geometry*, 34(2):102–115, 2006.

15 Subir Kumar Ghosh and Rolf Klein. Online algorithms for searching and exploration in the plane. *Computer Science Review*, 4(4):189–201, 2010.

16 Mikael Hammar, Bengt J. Nilsson, and Sven Schuierer. Parallel searching on m rays. *Computational Geometry*, 18(3):125–139, 2001.

17 Christoph A. Hipke, Christian Icking, Rolf Klein, and Elmar Langetepe. How to Find a Point on a Line Within a Fixed Distance. *Discrete Applied Mathematics*, 93(1):67–73, 1999.

18 Frank Hoffmann, Christian Icking, Rolf Klein, and Klaus Kriegel. The Polygon Exploration Problem. *SIAM Journal on Computing*, 31(2):577–600, 2001.

19 Christian Icking, Rolf Klein, Elmar Langetepe, Sven Schuierer, and Ines Semrau. An optimal competitive strategy for walking in streets. *SIAM Journal on Computing*, 33(2):462–486, 2004.

20 J.R. Isbell. An optimal search pattern. *Naval Research Logistics Quarterly*, 4(4):357–359, 1957.

21 Bala Kalyanasundaram and Kirk Pruhs. A Competitive Analysis of Algorithms for Searching Unknown Scenes. *Computational Geometry*, 3:139–155, 1993.

22 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):63–79, 1996.

23 Alejandro López-Ortiz and Sven Schuierer. The ultimate strategy to search on m rays? *Theoretical Computer Science*, 261(2):267–295, 2001.

24 Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

25 Marc van Kreveld. Competitive Analysis of the Pokémon Go Search Problem. In *Abstracts of the 33rd European Workshop on Computational Geometry*, pages 25–28, 2017. http://csconferences.mah.se/eurocg2017/proceedings.pdf.