Going Home Through an Unknown Street*

Alejandro López-Ortiz† Sven Schuierer‡

Abstract

We consider the problem of a robot traversing an unknown polygon with the aid of standard visibility. The robot has to find a path from a starting point s to a target point t. We provide upper and lower bounds on the ratio of the distance traveled by the robot in comparison to the length of a shortest path. Since this competitive ratio is unbounded for general polygons, we restrict ourselves to the well investigated class of *streets*. A *street* is a polygon where the part of the polygon boundary from s to t is weakly visible to the part from t to s and vice versa.

We consider two problems in this context. First we assume that the location of the target t is known to the robot. We prove a lower bound of $\sqrt{2}$ on the competitive ratio of any deterministic algorithm that solves this problem. This bound matches the competitive ratio for searches in a rectilinear polygon with an unknown destination which implies that, for rectilinear streets, knowledge of the location of the destination provides no advantage for the robot. In addition, we also obtain a lower bound of 9 for the competitive ratio of searching in generalized streets with known target which closely matches the upper bound if the target is unknown.

Secondly, we consider a strategy for searching in an arbitrarily oriented street where the location of t is unknown. We show that our strategy achieves a competitive ratio of $\sqrt{5}$ which significantly improves the best previously known ratio of $2\sqrt{1+1/\sqrt{2}}$.

1 Introduction

One of the main problems in robotics is to find a path from the current location of the robot to a given target. While most of the work in this area has focussed on efficient algorithms for path planning if the robot is given a map of its environment in advance, a more natural and realistic setting is to assume that the robot has only a partial knowledge of its surroundings.

In this paper we assume that the robot is equipped with a vision system that provides a visibility map of its *local* environment. Based on this information the robot has to find a path to a given target that is located somewhere within the scene. The search of the robot can be viewed as an

*This research is supported by the DFG-Project "Diskrete Probleme", No. Ot 64/8-1.

†Department of Computer Science, University of Waterloo, Waterloo, Ontario CANADA N2L 3G1,
e-mail: alopez-o@neumann.UWaterloo.ca

‡Department of Computer Science, University of Western Ontario, London, Ont., Canada N6A 5B7, and
Institut für Informatik, Universität Freiburg, Am Flughafen 17, D-79110 Freiburg, FRG, e-mail: schuiere@informatik.uni-freiburg.de

on-line problem since it discovers its surroundings as it travels. Hence, one way to analyze the quality of a search strategy is to use the framework of competitive analysis as introduced by Sleator and Tarjan [11]. A search strategy is called *c-competitive* if the path traveled by the robot to find the target is a most c times longer than a shortest path. c is called the *competitive ratio* of the strategy.

Since there is no strategy with a competitive ratio of $o(n)$ for scenes with arbitrary obstacles having a total of n vertices [2], the on-line search problem has been studied previously in various contexts where the geometry of the obstacles is restricted. Papadimitrou and Yannakakis were the first to consider the case of traversing an unknown scene with rectangular obstacles in search of a target whose location is known [10]. They show a lower bound of $\Omega(\sqrt{n})$ for the competitive ratio of any strategy. Later Blum, Raghavan, and Schieber provided a strategy that achieves this bound [2]. If the aspect ratio or the length of the longest side of the rectangles are bounded, better strategies are possible [3, 9].

Kleinberg studied the problem of a robot searching inside a simple polygon for an unknown goal located on the boundary of the polygon [8]. He introduced the notion of *essential cuts* inside a polygon of which there may be considerably fewer than polygon vertices and gave an $O(m)$-competitive strategy for polygons with m essential cuts.

Klein introduced the notion of a *street* which allowed for the first time a search strategy with constant competitive ratio even though the location of the target is unknown [7]. In a street the starting point s and the target t are located on the boundary of the polygon and the two polygonal chains from s to t are mutally weakly visible. Klein presents a strategy for searching in streets with a competitive ratio of $1 + 3/2\pi$. Though Klein's strategy performs well in practice—he reports that no example had been found for which his strategy performs worse than 1.8—the strategy and its analysis are both quite involved and no better competitive ratio could be shown until, recently, Kleinberg presented a new approach. His strategy for searching in streets allows to prove a competitive ratio of $2\sqrt{2}$ with a very simple analysis [8]. Moreover, for rectilinear streets Kleinberg shows that his strategy achieves a competitive ratio of $\sqrt{2}$ which is optimal due to the trivial example shown in Figure 1. If a strategy moves to the left or right before seeing t, then t can be placed on the opposite side, thus forcing the robot to travel more than $\sqrt{2}$ times the diagonal.

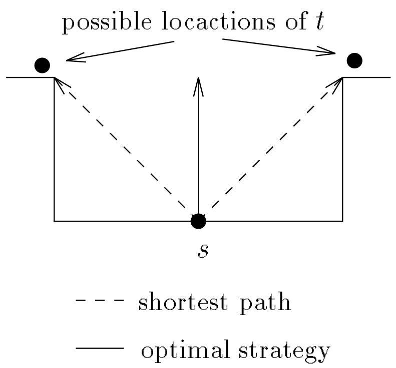

Figure 1: A lower bound for searching in rectilinear streets.

Curiously enough, this is the only known lower bound even for arbitrarily oriented streets. A different strategy with a competitive ratio of ~ 4.8 was developed by Icking [6]. Finally, a more general class of polygons, called $\mathcal{G}$-streets, has been introduced by Datta and Icking that allows search strategies with a competitive ratio of 9.06 [4]. All the above strategies fall into the category of Unknown Destination Searches (UDS) in which the location of the goal is unknown.

One natural source of information for the robot are the coordinates of the target. The first problem we consider is a lower bound for strategies for Known Destination Searches (KDS) in a street where the location of the goal is given in advance to the robot. In this case the example of Figure 1 obviously provides no longer a lower bound. We prove that in even in orthogonal streets a $\sqrt{2}$-competitive ratio is optimal as well, thus providing the first non-trivial lower bound for searching in polygons. This result is different from the general search problem as considered by Papadimitrou and Yannakakis in which knowledge of the destination improves the competitive ratio.

Secondly we consider a new strategy for searching in arbitrarily oriented streets. We achieve a competitive ratio of $\sqrt{5}$, providing the best performance guarantee for searching strategies in streets known so far.

The paper is organized as follows. In Section 2 we introduce the basic geometric concepts necessary for the rest of the paper. In particular, we give a precise definition of a street. In Section 3 we show that any deterministic search algorithm for orthogonal streets that knows the location of the target can be forced to travel $2 - O(1/\sqrt{n})$ times the distance of a shortest path to the target where n is the number of vertices of the polygon. Finally, Section 4 deals with a new strategy to search in streets and its analysis.

2 Definitions and Assumptions

We consider a simple polygon P in the plane with n vertices and a robot inside P which is located at a start point s on the boundary of P. The robot has to find a path from s to the target point t. The search of the robot is aided by simple vision (i.e. we assume that the robot knows the visibility polygon of its current location). Furthermore, the robot retains all the information seen so far (in memory) and knows its starting and current position. We are, in particular, concerned with a special class of polygons called *streets* first introduced by Klein [7].

Definition 2.1 [7] *Let P be a simple polygon with two distinguished vertices, s and t, and let L and R denote the clockwise and counterclockwise, resp., oriented boundary chains leading from s to t. If L and R are mutually weakly visible, i.e. if each point of L sees at least one point of R and vice versa, then (P, s, t) is called a* street.

Definition 2.2 *In the class of Known Destination Search (KDS) problems, a robot searches a simple rectilinear polygon, starting from s on the boundary of the polygon, for a target point t on the boundary of P with known location.*

We denote the L_2-distance between two points p_1 and p_2 by $d(p_1, p_2)$ and the L_2-norm of a point p by $\|p\|$.

Definition 2.3 *Let P be a street with start point s and target t. If p is a point of P, then the* visibility polygon *of p is the set of all points in P that are seen by p. It is denoted by $V(p)$. A* window *of $V(p)$ is an edge of $V(p)$ that does not belong to the boundary of P (see Figure 2).*

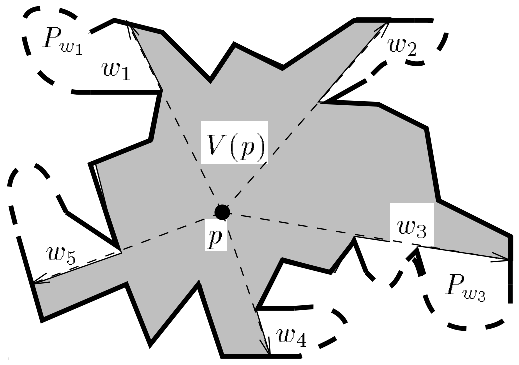

Figure 2: The the visibility polygon $V(p)$ of p with windows $w_1, \ldots, w_5$.

A window w splits P into a number of subpolygons $P_1, \ldots, P_k$ one of which contains $V(p)$. We denote the union of the subpolygons that do not contain $V(p)$ by P_w.

3 A $\sqrt{2}$-competitive lower bound

We construct a family of polygons which are $(\sqrt{2} - \epsilon)$-competitive for KDS for any $\epsilon > 0$. First, we define some widgets which will be used in the general construction.

Definition 3.1 *An* eared rectangle *is a rectangle two units wide and one unit tall. The center of the base is the* entry point *and on the top left and right corners there are two small alleys (ears) attached to it (see Figure 3). One of the alleys is* connecting, *the other is* a dead alley.

Definition 3.2 *The* aspect ratio A *of a general polygon is defined as the ratio between the smallest and the longest edge of the polygon. Thus* $A \leq 1$.

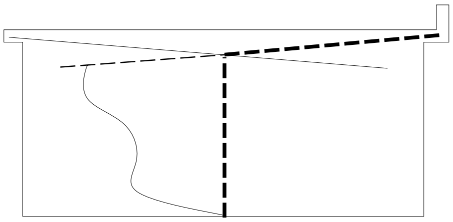

Figure 3: Eared Rectangle, with walk inside.

Lemma 3.1 *An eared rectangle may be traversed from the entry point to the connecting alley at a* $(\sqrt{2} - \epsilon)$*-competitive ratio, with* $\epsilon = O(A)$*, which is optimal.*

Proof: First we show that a $\sqrt{2}$-competitive ratio is attainable. The robot walks up the middle of the rectangle, until it sees the top boundaries of both alleys. At this time the robot can see into either alley and determine which one is open, and proceed to walk in this direction (see bold dashed lines in Figure 3). The length of the trajectory is $1 - tan\ \theta + 1/cos\ \theta$, where θ is the angle of the line between the extreme upper and the closer lower end point of the alleys. Notice that θ can be made as small as needed by means of reducing the height of the alley. Thus, this strategy gives a walk of length arbitrarily close to $\sup_{\theta \to 0}\{1 + 1/cos\ \theta - tan\ \theta\} = 2$. The optimal walk is of length $\sqrt{2}$ for a competitive ratio of $\sqrt{2} - \epsilon$ where $\epsilon = (1/\sqrt{2})(1 + tan\ \theta - 1/cos\ \theta) = O(A)$.

This strategy is optimal as well. We use an adversary argument to show this. The adversary simply opens the first alley to be looked into by the robot, and closes the other alley. Clearly the alley opened is always in the opposite half of the rectangle in which the robot is currently located. (see curvy path plus dashed line

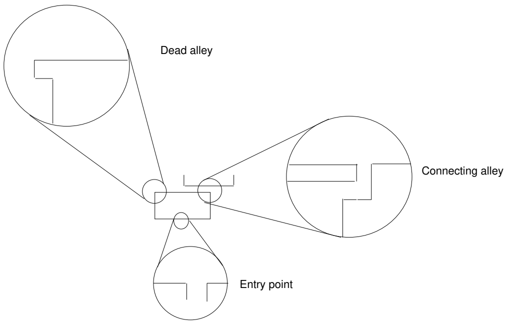

Figure 4: Interconnecting Eared Rectangles.

in Figure 3). A simple application of the triangle inequality shows that the path in bold is shorter, and thus has a better competitive ratio.

□

Eared rectangles can be interconnected to create paths. Figure 4 shows the details of such rectangles.

Theorem 1 *There exists a street with n vertices which can be searched with an optimal competitive ratio of $\sqrt{2} - O(1/\sqrt{n})$.*

Proof: As proved by Kleinberg [8], there exists a $\sqrt{2}$-competitive strategy for UDS which can be applied in a straightforward way to the KDS problem and gives a strategy of the same competitive ratio for all polygons in the KDS problem.

What remains to be shown is that this competitive ratio is optimal. We assume that the target is a some distance directly above the start point as shown in Figure 5. To prove a lower bound of $\sqrt{2}$ we first consider two extreme cases of interconnecting eared rectangles, namely the *Walk the Middle Policy* and the *Always to the Right Policy*.

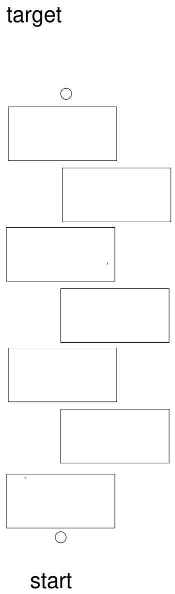

Figure 5: Walk the Middle Policy

If the algorithm uses a strategy such as the one proposed in Lemma 3.1, the construction of Figure 5 shows an example of a polygon with a competitive ratio of $(2n+1)/(\sqrt{2}n+1)$, where n is the number of rectangles between the start point and the target.

Thus, an algorithm needs to deviate from the *Walk the Middle Policy.* In this case, the adversary presents the algorithm with an eared rectangle and it opens and closes the alleys according to the strategy proposed in Lemma 3.1. If we assume then that the algorithm always meets the line of sight in the left half, the adversary then consistently opens the right alley (see Figure 6). This creates a staircase moving to the right. Notice that the L_1 distance from the current robot position to the target is always within one unit of the L_1 distance from the start point to the target. That is, the adversary has forced the algorithm

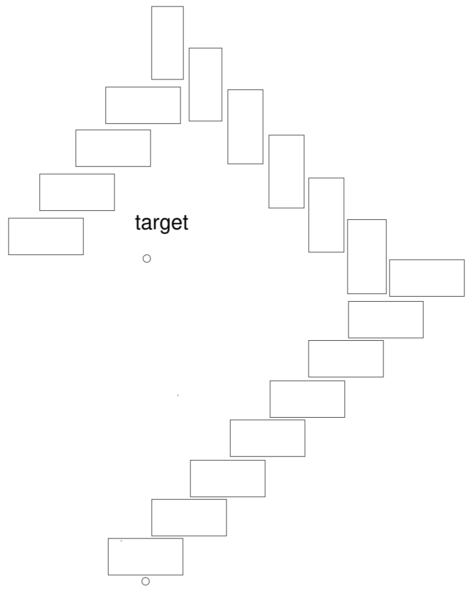

Figure 6: Always to the Right Policy

Figure 7: Wavering Policy

to move at a worse than $\sqrt{2}$ competitive ratio, but the target is no closer than before.

When the current connecting alley is now horizontally aligned with the target, the adversary moves one unit closer to the target (we assume that the algorithm also moves optimally in this part, since it knows the position of the target) and proceeds to construct a new staircase. This results in a spiraling set of staircases converging to the start point. The spiral is of length quadratic in n (see [1]) and, thus, the competitive ratio is $O((2n^2 + n)/(\sqrt{2}n^2+n))$ which goes to $\sqrt{2}$ as n goes to infinity.

Having analyzed these extreme cases, we now consider a *Wavering Policy* in which the algorithm neither walks up the middle, nor consistently slants either way (see Figure 7). The *Walk the middle Policy* and *Always to the right Policy* can be viewed as extreme instances of the *Wavering Policy.*

In the case of a wavering algorithm, the adversary maintains the strategy described above. Every time the algorithm deviates from the *Always to the Right Policy*, the adversary moves to the left. As a consequence, the L_1 distance to the target is reduced by two units, while the competitive ratio remains above $\sqrt{2}$.

From the point of view of the algorithm a left turn, is equivalent to a "jump" from one level of the spiral to a level on the spiral associated to a start point two units closer to the target (see Figure 7, with the polygon in solid lines, and the older staircase in dashed lines).

Let k be the number of turns to the left deviating from the *Always to the Right Policy.* Without loss of generality let $k \leq m$, since the the case $k > m$ can be seen as a deviation from the symmetric *Always to the Left Policy.* Furthermore, assume that the algorithm jumps at staircases $a_1, a_2, \ldots, a_k$, where the staircases are numbered, starting from 1, in the order they are traversed. Then, the total length of the path traversed by the algorithm is

$$\begin{aligned}
&2 \sum_{0 \le j < a_1} (n - 2j) + 2 + \\
&2 \sum_{0 \le j < a_2} (n - 2a_1 - 2j) + 2 + \\
&2 \sum_{0 \le j < a_3} (n - 2a_1 - 2a_2 - 2j) + 2 + \cdots
\end{aligned}$$

Lemma 3.2 *Consider two strategies for walking up the staircase. Strategy A turns left in staircases $\{a_i\}_{1 \le i \le k}$, and Strategy B turns left in the staircases $\{b_i\}_{1 \le i \le k}$, such that $b_i = a_i - 1$, for all i with $a_i > 1$, and $b_i = 1$ otherwise. Then strategy B has a better competitive ratio than A.*

Proof: Since $a_i \ge b_i$ it follows that the summation above is, term by term, larger for strategy A than for strategy B, from which the claim follows. $\square$

Thus, setting $a_i = 1$, for all i, is optimal. Let $n = 2m$. If the algorithm jumps or turns left k times, then we have for $l = m - k$

Length of shortened spiral

$$= \quad n + \sum_{i=0}^{l} (2l - 2i) = n + l(l+1)$$

Length of optimal walk =

$$\sqrt{2}\,(n + l(l+1)) + n - 2k$$

Distance traversed by the algorithm =

$$2\,(n + l(l+1)) + n - 2k$$

Competitive ratio $= \sqrt{2} - \xi$ where

$$\xi = \frac{2(\sqrt{2}-1)l}{\sqrt{2}\,(n + l(l+1)) + n - 2k}.$$

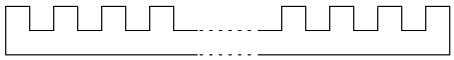

Figure 8: A "rake" polygon.

To improve its competitive ratio, the algorithm can select the optimal value of k for all given m that maximizes ξ. As proven in Appendix 1, $k = m - \sqrt{2m}$ maximizes ξ to $O(1/\sqrt{m})$. Since each eared rectangle is traversed at a $\sqrt{2} - O(A)$ competitive ratio, we have that the adversary strategy described above forces any algorithm into a $\sqrt{2} - O(A) - O(1/\sqrt{n})$ competitive ratio, which in the limit is $\sqrt{2}$.

As it can be seen, regardless of the policy, a $\sqrt{2}$ inefficiency factor is necessarily introduced, even in the case where the robot knows where it is going, but is ignorant of the terrain in which is moving. $\square$

Datta and Icking [4] introduced the notion of $\mathcal{G}$-streets and showed for the UDS problem a competitive ratio of 9.06.

Definition 3.3 **([4])** *A simple polygon in the plane is called a* generalized street *if for every boundary point $p \in L \cup R$, there exists a horizontal chord with end points in L and R and from which p is weakly visible.*

Datta and Icking proved a lower bound by building a "rake" polygon which is traversed at a 9-competitive ratio on the limit (see Figure 8).

For the KDS problem, as in the case of eared rectangles, a single rake polygon does not suffice to obtain a lower bound on the competitive ratio of 9.

A similar strategy to the one presented results in a 9-competitive ratio for this problem.

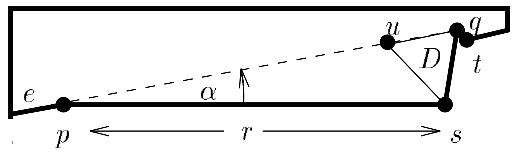

Figure 9: A lower bound example for Kleinberg's strategy.

Corollary 1 *There exists a family of orthogonal $\mathcal{G}$-streets with 9-competitive ratio, which is close to optimal.*

Proof: Consider a target at a distance n of the start point, and placed directly above it on the vertical direction. Then a set of n^2 rakes placed one above the other, each of height $1/n$, gives the desired lower bound. □

4 Traversing a Street

In this section we present a new strategy to traverse a street from the starting point s to the target point t. Our strategy closely follows the approach of Kleinberg [8] in order to deal with the simple cases which his strategy handles optimally but deviates in the more complicated cases.

The competitive ratio of Kleinberg's strategy is shown to be $2\sqrt{2}$ ($\sim$ 2.83) in his paper but a tighter analysis—also mentioned in his paper—yields an upper bound of $2\sqrt{1+1/\sqrt{2}}$ ($\sim$ 2.61). Before we describe our approach we show that this bound is tight for his strategy. Figure 9 shows an example of a polygon where his bound is achieved asymptotically.

Here Kleinberg's strategy follows the diagonal D to point u where the chain to the left of point p is visible to the robot and then moves to the right to point q. In order to see that a competitive ratio of $2\sqrt{1+1/\sqrt{2}}$ is achieved let $d = \sqrt{2}+1$ and q be chosen to have the coordinates $(q_1, q_2) = (1/\sqrt{d^2+1}, d/\sqrt{d^2+1})$ where we assume s to be the origin. If p has distance r to s and we observe that $\sin\alpha = q_1/(r+q_2)$, then $u = (u_1, u_2) = (r\sin\alpha/(\sin\alpha - 1), r\sin\alpha/(\sin\alpha - 1))$. Hence, the total distance traveled by a robot following Kleinberg's strategy is $\geq \sqrt{2}u_2 + u_1 + q_1$. If we take into account that $u_1 = u_2 = q_2 - O(1/r)$, then the distance traveled by the robot is $q_1 + (\sqrt{2}+1)q_2 - O(1/r) = 1/\sqrt{d^2+1} + d^2/\sqrt{d^2+1} - O(1/r) = \sqrt{d^2+1} - O(1/r)$ which tends to $2\sqrt{1+1/\sqrt{2}}$ as r approaches $+\infty$. By keeping edge e collinear with q and moving t closer and closer to q it can be seen that the claimed ratio can be achieved arbitrarily closely.

Our strategy can be shown to have a competitive ratio of at most $\sqrt{5}$ ($\sim$ 2.237) but in contrast to Kleinberg's analysis, the analysis of our strategy is not tight. Before we describe the strategy we need a few definitions and observations.

As observed by Kleinberg the shortest path $\mathcal{P}$ from s to t consists of a number of line segments that touch reflex vertices of P. The general strategy we follow is to start at a reflex vertex v of P that belongs to $\mathcal{P}$ and to identify another reflex vertex v' of $\mathcal{P}$ that is closer to t by traveling further on. If the robot has identified v', then it moves to it and starts the search anew. A move from one reflex vertex of P on $\mathcal{P}$ to another closer to t is called a *step*.

Recall that a window is an edge of the boundary of the visibility polygon $V(p)$ of p that does not belong to the boundary of P.

All windows are collinear with p. The end point of a window w that is closer to

p is called the *entrance point* of w. We assume that a window w has the orientation of the ray from p to entrance point of w. We say a window w is a *left window* if the part P_w of P that does not contain $V(p)$ is locally to the left of w w.r.t. the given orientation of w. A *right window* is defined similarly.

If the robot has traveled along the path $\mathcal{P}$, then we assume that the robot knows the part of P that can be seen from $\mathcal{P}$, i.e. the robot has seen maintains the polygon $V(\mathcal{P}) = \bigcup_{p \in \mathcal{P}} V(p)$. We say a window w of $V(p)$ is a *true window* w.r.t. $\mathcal{P}$ if it is also a window of $V(\mathcal{P})$.

We have the following lemmas about true windows which are proved by Klein with a slightly different terminology [7].

Lemma 4.1 *If w is a right (left) window of $V(p)$ and the boundary of P_w belongs to L (R), then w is not a true window.*

We say two windows w_1 and w_2 are *clockwise consecutive* if the clockwise oriented polygonal chain of $V(p)$ between w_1 and w_2 does not contain a window different from w_1 and w_2. *Counterclockwise consecutive* is defined analogously.

Lemma 4.2 *All windows that belong to L (R) are clockwise (counterclockwise) consecutive in $V(p)$.*

True windows are called *consecutive* if there is no true window that is between them. An immediate corollary of Lemmas 4.1 and 4.2 is that true left and true right windows are consecutive.

Corollary 4.3 *If w_0 is the window that is intersected by $\mathcal{P}$ the first time, then all true left (right) windows are clockwise (counterclockwise) consecutive from w_0 in $V(p)$.*

Because of Corollary 4.3 there is a clockwise-most true left entrance point from w_0 which we denote by p^+ and a counterclockwise-most true right entrance point of $V(p)$ which we denote by p^- if $V(p)$ contains both true left and right windows. The point p^+ is called the *left extreme entrance point* and p^- the *right extreme entrance point* of $V(p)$. Klein shows that the shortest path from s to t goes either through the left or right extreme entrance point [7]. For further reference we state this result as a lemma.

Lemma 4.4 *The shortest path from s to t goes either through the left or right extreme entrance point.*

Now assume the robot starts at s and travels towards its target. We consider five cases:

Case 1 t is visible to the robot.
The robot moves to t on a straight line.

Case 2 There is no true left window (right window).
The robot moves to p^- (p^+).

Case 3 The angle at the location of the robot between p^+ and p^- greater equal $90°$.
We apply Algorithm *Move-in-Quadrant* as described below until we are able to decide which of p^+ and p^- is part of a shortest path from s to t.

If none of the above cases apply, then p^+ and p^- are defined and the angle at p between p^+ and p^- is less than $90°$. The robot chooses a direction such that p^+ is to the left of the direction and p^- to the right. It travels following the direction until one of the above or one of the following two cases occurs.

Case 4 A new point p^+ or p^- appears and p, p^+ and p^- are collinear.
The robot moves along the line through p, p^+, and p^- to the closer point of p^+ and p^-.

Case 5 The angle at the location of the robot between p^+ and p^- equals $90°$.
We apply Algorithm *Move-in-Quadrant* as described below until we are able to decide which of p^+ and p^- is part of a shortest path from s to t.

The *orthogonal projection* p' of a point p onto a line segment l is defined as the point of l that is closest to p. If the line segment joining p with p' is not orthogonal to l, then p' is a *degenerate* orthogonal projection.

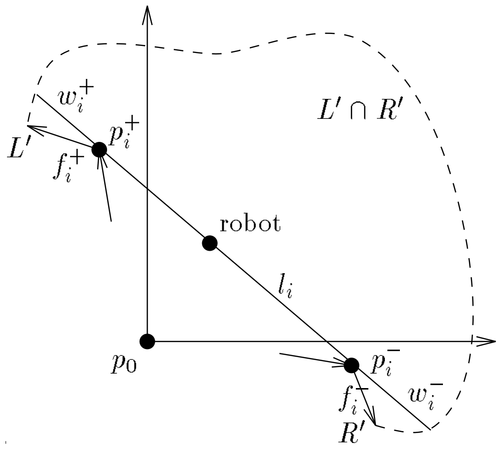

Figure 10: If the robot has reached l_i, then one of p_i^+ or p_i^- is not an extreme entrance point any more.

Algorithm Move-in-Quadrant

Input: A point p_0 in P such that the angle at p_0 of p_0^+ and p_0^- of $V(p_0)$ is $\geq 90°$;

$i := 0$;
while p_i^+ and p_i^- are defined **do**
(1) Move to the orthogonal projection p_{i+1} of p_0 onto the line segment l_i from p_i^+ to p_i^-;
Compute the points p_{i+1}^+ and p_{i+1}^- of $V(p_{i+1})$;
$i := i + 1$;
end while;

The correctness of the algorithm follows from the following lemma.

Lemma 4.5 *If the robot has reached the line segment l_i, then one of p_i^+ or p_i^- is not an extreme entrance point of $V(p_{i+1})$ anymore.*

Proof: The proof is by contradiction. Let f_i^+ be the outgoing edge at p_i^+ given a *counterclockwise* orientation of the boundary of P. Similarly, let f_i^- be the outgoing edge at p_i^- given a *clockwise* orientation of the boundary of P (see Figure 10).

Assume that the robot has reached l_i and there are a left and a right extreme true window w_i^+ and w_i^- incident to p_i^+ and p_i^-, respectively. The target t is either to the left of w_i^+ or the right of w_i^-. Hence, either the counterclockwise oriented polygon boundary L' from p_i^+ to the upper end point of w_i^- or the clockwise oriented polygon boundary R' from p_i^- to the upper end point of w_i^+ belongs completely to one chain. Since f_i^+ belongs to and sees only L' and f_i^- belongs to and sees on R', one of the two edges sees only edges of the chain it belongs to in contradiction to the assumption that P is a street. $\square$

For the analysis consider the Cases 1, 2, and 4 first. In the Cases 1 and 2 the robot moves directly to the next point on a shortest path from s to t, hence, the competitive ratio is 1. If Case 4 occurs before

Algorithm *Move-in-Quadrant* is invoked, then the angle between the line segment from the robot to p_i^+ or p_i^- and the traveling direction is less or equal 90° which implies that if the robot moves directly to p_i^+ or p_i^-, then the competitive ratio is bounded by $\sqrt{2}$.

4.1 Analysis of the Algorithm *Move-in-Quadrant*

If during the movement in Step (1) one of the Cases 1 or 4 occurs, then the robot moves immediately to t or the closer of the points p^+ or p^-. Nevertheless, for the analysis we assume that the robot first moves to the line segment determined by the old points p^+ and p^- and then to the closer one of the two.

In the following we assume that the Algorithm *Move-in-Quadrant* has stopped after k iterations. We obtain a simple corollary to Lemma 4.4.

Corollary 4.6 *During the Algorithm Move-in-Quadrant the shortest path from s to t goes through either p_i^+ or p_i^-, for all $0 \leq i \leq k$.*

Because of Corollary 4.6 it suffices to bound the ratio of the length of the path traveled by the robot to the distance between p_0 and p_i^+ or p_i^-—whichever is detected as a part of the shortest path from s to t.

Our approach to analyze our strategy is based on the simple idea of incremental accounting, that is, we compare the increase in distance to p_0 in Iteration i which is $d(p_0, p_{i+1}) - d(p_0, p_i)$ to the distance traveled by the robot which is $d(p_i, p_{i+1})$.

The situation we analyze is displayed in Figure 11. We introduce a coordinate system where p_0 is the origin and the x-axis passes through p_0^-. Note that since the angle at p_0 between between p_0^+ and p_0^- is greater or equal 90°, there is no reflex vertex of P in the first quadrat of the introduced coordinate system that is visible to p_0. Assume that we have arrived at point p_i and move to point p_{i+1} in the next iteration. We make a few simple observations about the locations of p_i^+, p_i^-, and the line segment l_i from p_i^+ to p_i^-.

Lemma 4.7 *The points p_i^+ belong to the second quadrant and the point p_i^- belong to the fourth quadrant, for all $0 \leq i \leq k-1$.*

Proof: The proof is by induction. For $i = 0$ the claim clearly holds. Let l be the directed line from p_i^+ to p_i^-. The points p_{i+1}^+ and p_{i+1}^- both lie in the halfplane to the left of l and, hence, they are not contained in the third quadrant. If p_{i+1}^+ is contained in the first quadrant, then p_0^+ is not a left extreme entrance point. If p_{i+1}^+ is contained in the fourth quadrant, then there is a point on the path of the robot where p_{i+1}^- and p_{i+1}^+ are collinear with the robot position and the robot stops with $k = i + 1$. □

Lemma 4.8 *The line segments l_i and l_{i+1} do not cross in the first quadrant.*

Proof: Let l be the directed line from p_i^+ to p_i^-. Since the points p_{i+1}^+ and p_{i+1}^- both lie in the halfplane to the left of l, l_{i+1} does not cross l and, therefore, l_i. □

Since the line segment l_i intersects the first, second, and fourth quadrant, the orthogonal projection of p_0 onto l_i is nondegenerate.

Let L_i be the length of the path $\mathcal{P}_i$ traveled by the robot to reach p_i and r_i the distance of p_i to p_0. It is our aim to show that $L_i \leq 2r_i$, for all $0 \leq i \leq k$. In order to simplify the analysis we consider the line

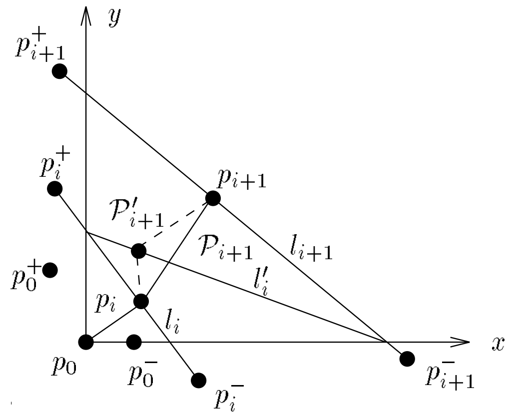

Figure 11: Introducing a new segment between l_i and l_{i+1}.

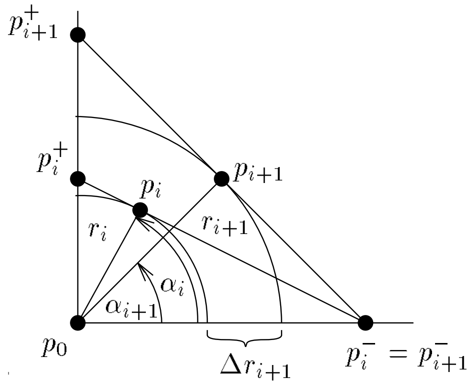

Figure 12: Bounding $\Delta L_{i+1} + \Delta Q_{i+1}$ by Δr_{i+1}.

segment l'_i from the intersection point of l_i with the y-axis to the intersection point of l_{i+1} with the x-axis as shown Figure 11.

The line segment l'_i is located between l_i and l_{i+1}. If we consider the path $\mathcal{P}'_i$ from p_0 to p_i that visits the orthogonal projections of p_0 onto the line segments l_j and l'_j in order, for $0 \leq j \leq i$, then the length L'_i of $\mathcal{P}'_i$ is obviously greater than or equal to L_i. Since $\mathcal{P}_i$ and $\mathcal{P}'_i$ start and end at the same point, we can bound the length of $\mathcal{P}_i$ w.r.t. r_i if we can bound the length of $\mathcal{P}'_i$ w.r.t. r_i. Hence, we can assume in the following that p_i^+ is located on the y-axis, p_i^- on the x-axis, and either $p_i^+ = p_{i+1}^+$ or $p_i^- = p_{i+1}^-$.

So suppose the robot has reached the point p_i at distance r_i to p_0 and $L_i \leq 2r_i$. Note that the line segment l_i is the tangent at p_i to the circle with radius r_i and center p_0 (see Figure 12.

The robot moves now from p_i to p_{i+1} at distance r_{i+1} to p_0. Again the line segment l_{i+1} is the tangent at p_{i+1} to the circle with radius r_{i+1} and center p_0. Let α_i be the angle of the line through p_0 and p_i formed with the x-axis. Let Δr_{i+1} be the difference between r_{i+1} and r_i and ΔL_{i+1} the difference between L_{i+1} and L_i. If we could bound ΔL_{i+1} by $2\Delta r_{i+1}$, then we immediately would have $L_{i+1} \leq 2r_{i+1}$. Unfortunately, the ratio $\Delta L_{i+1}/\Delta r_{i+1}$ is unbounded if p_i and p_{i+1} form a very small angle with the x-axis. Furthermore, if $\alpha_i = \pi/4$ and $\alpha_{i+1} = 0$, then $\Delta L_{i+1}/\Delta r_{i+1} \geq 2.41$. Therefore, we introduce a potential function Q that assigns the value $\frac{1}{2} r_i \sin(2\alpha_i)$ to each point p_i. The intuition behind this potential function is that if we are close to $\pi/4$, then L_i is also close to r_i. Let $\Delta Q_{i+1} = Q(p_{i+1}) - Q(p_i)$. In the following we show that $\Delta L_{i+1} + \Delta Q_{i+1} \leq 2\Delta r_{i+1}$.

We assume that $\alpha_i \geq \alpha_{i+1}$. If $\alpha_i < \alpha_{i+1}$, then we mirror p_i and p_{i+1} at the diagonal of the first quadrant. This operation does not change ΔL_{i+1} and Δr_{i+1}. The angle α_i is mirrored to $\alpha'_i = \pi/2 - \alpha_i$. Since $\frac{1}{2} r_i \sin(2\alpha'_i) = \frac{1}{2} r_i \sin(\pi - 2\alpha_i) = \frac{1}{2} r_i \sin(2\alpha_i)$, ΔQ_{i+1} remains unchanged as well and the analysis for $\alpha_i \geq \alpha_{i+1}$ applies.

If $\alpha_i \geq \alpha_{i+1}$, then $p_{i+1}^- = p_i^-$ and $r_{i+1} =$

$(\cos\alpha_{i+1}/\cos\alpha_i)r_i$. Hence, we obtain

$$\begin{aligned}\Delta r_{i+1} &= \frac{\cos\alpha_{i+1}}{\cos\alpha_i} r_i - r_i \\ &= \frac{r_i}{\cos\alpha_i}(\cos\alpha_{i+1} - \cos\alpha_i)\end{aligned}$$

and

$$\begin{aligned}\Delta Q_{i+1} &= \frac{1}{2} r_{i+1}\sin(2\alpha_{i+1}) - \frac{1}{2} r_i \sin(2\alpha_i) \\ &= \frac{1}{2}\frac{r_i}{\cos\alpha_i}(\sin(2\alpha_{i+1})\cos\alpha_{i+1} - \\ &\qquad \sin(2\alpha_i)\cos\alpha_i)\end{aligned}$$

In order to compute ΔL_{i+1} we first observe that the coordinates of p_i are given by $r_i(\cos\alpha_i, \sin\alpha_i)$ and the coordinates of p_{i+1} by $r_{i+1}(\cos\alpha_{i+1}, \sin\alpha_{i+1})$. Therefore, ΔL_{i+1} equals

$$\left\| r_i \frac{\cos\alpha_{i+1}}{\cos\alpha_i}\begin{pmatrix}\cos\alpha_{i+1}\\ \sin\alpha_{i+1}\end{pmatrix} - r_i\begin{pmatrix}\cos\alpha_i \\ \sin\alpha_i\end{pmatrix}\right\|.$$

After some algebraic manipulations we obtain the following formula for ΔL_{i+}

$$\frac{r_i}{\cos\alpha_i}\sqrt{1/2(1-\cos(2\alpha_{i+1}-2\alpha_i))}.$$

In order to prove that $\Delta L_{i+1} + \Delta Q_{i+1} \leq 2\Delta r_{i+1}$, we consider the function $f(\alpha_i, \alpha_{i+1})$ which is defined as

$$2\frac{\cos\alpha_i}{r_i}(2\Delta r_{i+1} - \Delta L_{i+1} - \Delta Q_{i+1}).$$

If we can prove that $f(\alpha_i, \alpha_{i+1}) \geq 0$, for all $0 \leq \alpha_{i+1} \leq \alpha_i \leq \pi/2$, then the claim follows. We show that $\partial f/\partial\alpha_i \geq 0$, for all $0 \leq \alpha_{i+1} \leq \alpha_i \leq \pi/2$. To this end we split f into the two functions

$$g_1(\alpha_i, \alpha_{i+1}) = 2\frac{\cos\alpha_i}{r_i}(2\Delta r_{i+1} - \Delta Q_{i+1})$$

and

$$g_2(\alpha_i, \alpha_{i+1}) = 2\frac{\cos\alpha_i}{r_i}\Delta L_{i+1}$$

with $f = g_1 - g_2$. We show that $\partial g_1/\partial\alpha_i \geq 2$ and $\partial g_2/\partial\alpha_i \leq 2$. This is again done by considering the derivatives. For g_1 the second derivative $\partial^2 g_1/(\partial\alpha_i)^2$ equals

$$4\sin\alpha_i + 2\cos(2\alpha_i)\cos\alpha_i - \sin(2\alpha_i)\sin\alpha_i.$$

The roots of the equation $\partial^2 g_1/(\partial\alpha_i)^2 = 0$ are given by $2\arctan x_j$, where x_j is a root of the polynomial $q(x) = x^6 + 7x^5 + x^4 - 22x^3 - x^2 + 7x - 1$. The polynomial q has six roots, three of which are negative and lead to a negative angle. One is in the interval $[0.1, 0.2]$, one in the interval $[0.4, 0.5]$, and one in the interval $[1.4, 1.5]$. Since $2\arctan(1.4) > 1.8 > \pi/2$, only the first two are of interest. If x_1 is the root of q in $[0.4, 0.5]$, then $\beta_1 = 2\arctan(x_1) \in [0.88, 0.89]$ and

$$\begin{aligned}\frac{\partial g_1}{\partial\alpha_i}(\beta_1) &\geq \min\left(\frac{\partial g_1}{\partial\alpha_i}(0.88), \frac{\partial g_1}{\partial\alpha_i}(0.89)\right) \\ &\qquad -7\cdot(0.89-0.88) \\ &\geq 2.086 - 0.07 \\ &> 2\end{aligned}$$

since $\partial^2 g_1/(\partial\alpha_i)^2 \leq 7$. Similarly, if x_2 is the root of q in $[0.15, 0.16]$, then $\beta_2 = 2\arctan(x_2) \in [0.29, 0.32]$ and

$$\frac{\partial g_1}{\partial\alpha_i}(\beta_1) \geq 2.59 - 7\cdot 0.03 > 2.$$

Finally, at the boundaries of the interval $[0, \pi/2]$ we have $\partial g_1/\partial\alpha_i(0) = 2$ and $\partial g_1/\partial\alpha_i(\pi/2) = 4$. Hence, $\partial g_1/\partial\alpha_i \geq 2$.

Now consider the derivative of g_2 which is $\partial g_2/\partial\alpha_i = 2\sin(2\alpha_{i+1} - 2\alpha_i)/\sqrt{2-2\cos(2\alpha_{i+1}-2\alpha_i)}$. We claim that $\partial g_2/\partial\alpha_i \leq 2$. This can be seen as follows.

$$\begin{aligned}\frac{-2\sin(2\alpha_{i+1}-2\alpha_i)}{\sqrt{2-2\cos(2\alpha_{i+1}-2\alpha_i)}} &\leq 2 \quad \Longleftarrow \\ \frac{4\sin^2(2\alpha_{i+1}-2\alpha_i)}{2-2\cos(2\alpha_{i+1}-2\alpha_i)} &\leq 4 \quad \Longleftrightarrow \\ (1-\cos(2\alpha_{i+1}-2\alpha_i))^2 &\geq 0\end{aligned}$$

Therefore, $\partial f/\partial\alpha_i = \partial g_1/\partial\alpha_i - \partial g_2/\partial\alpha_i \geq 2 - 2 = 0$ and f is monotone in α_i. The smallest value allowed for α_i is α_{i+1} since we assume that $\alpha_i \geq \alpha_{i+1}$. But $f(\alpha_{i+1}, \alpha_{i+1}) = 0$, for all $0 \leq \alpha_{i+1} \leq \pi/2$ and, hence, $f(\alpha_i, \alpha_{i+1}) \geq 0$, for all $0 \leq \alpha_{i+1} \leq \alpha_i \leq \pi/2$ which proves that $\Delta L_{i+1} + \Delta Q_{i+1} \leq 2\Delta r_{i+1}$.

Hence,

$$\begin{aligned} 2r_{i+1} &= \sum_{j=0}^{i} 2\Delta r_{j+1} \\ &\geq \sum_{j=0}^{i} \Delta L_{j+1} + \sum_{j=0}^{i} \Delta Q_{j+1} \\ &= L_{i+1} + Q(p_{i+1}) - Q(p_0) \end{aligned}$$

Since $Q(p_0) = \frac{1}{2} r_0 \sin\alpha_0 = 0$ and $Q(p_{i+1}) \geq 0$, the claim follows. In fact, we have shown that

$$L_{i+1} \leq (2 - \frac{1}{2}\sin\alpha_{i+1}) r_{i+1}. \quad (1)$$

4.2 Analysis of Cases 3 and 5

In order to obtain the final competitive ratio for one step we have to take into account that the robot has to move to either p^+_{k-1} or p^-_{k-1}. First consider Case 3. If p^-_k is undefined, then p^+_{k-1} belongs to the shortest path from s to t. We assume that the length of the line segment l'_{k-1} between p_0 and p^+_{k-1} is 1 and that the angle between the line segment l_k from p_0 to p_k and l'_{k-1} is β (see Figure 13). Since l_k is orthogonal to the line segment between p^+_{k-1} and p^-_{k-1}, the length of l_k is $\cos\beta$ and the distance from p_k to p^+_{k-1} is $\sin\beta$. Hence, the maximum distance traveled by the robot from p_0 to p^+_{k-1} is bounded by

$$\max_{0\leq\beta\leq\pi/2} \sin\beta + 2\cos\beta$$

or

$$\max_{0\leq x\leq 1} \sqrt{1-x^2} + 2x.$$

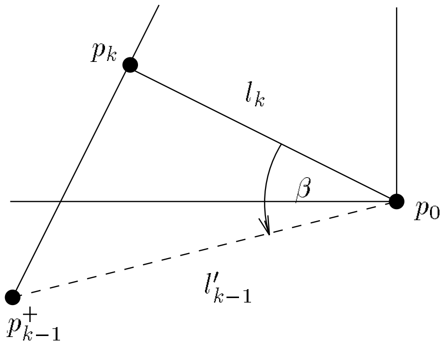

Figure 13: Bounding the competitive ratio in Case 3.

This maximum is achieved for $x = 2/\sqrt{5}$ and yields a value of $\sqrt{5}$. The same analysis applies if p^+_k is undefined.

Now we consider Case 5. This case turns out to be somewhat more complicated than the previous analysis. Let p be the point where the robot started its search. After traveling some distance a the robot encounters a point p_0 where the angle between p^+ and p^- is exactly 90°. Note that the angle between p^+ and p^- at the robot position is a continuous, monotonously increasing function if the robot moves on a ray between p^+ and p^-. At p_0 the robot invokes the Algorithm *Move-in-Quadrant*. First note that since there are no reflex vertices visible in the first quadrant of the coordinate system located at p_0, the x- and the y-coordinate of p are negative.

Suppose that p^-_k is undefined. Then, a shortest path from s to t visits p (by the induction hypothesis), p^+_1, and p^+_{k-1}. Hence, we have the situation displayed in Figure 14.

The length of the shortest path from p to p^+_{k-1} is at least the sum of the distances d from p to p^+_1 and e from p^+_1 to p^+_{k-1}. We first consider for which point p^+_1 the

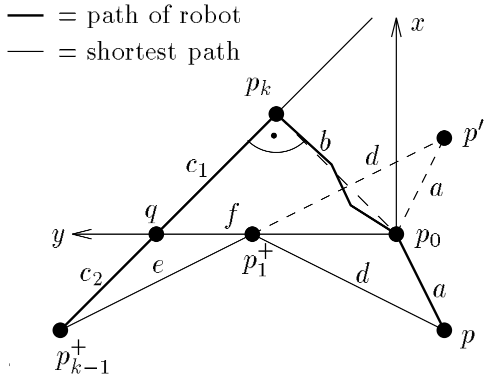

Figure 14: Computing the competitive ratio in Case 5.

sum $d + e$ is minimized given p, p_0, and p_{k-1}^+. In order to compute this point p_1^+ we reflect p at the y-axis to a point p' and note that the distance between p' and p_1^+ equals d (see Figure 14). Hence, $d + e$ is minimized if p_1^+ is located on the line from p_{k-1}^+ to p'. Furthermore, the distance a from p to p_0 which is traveled by the robot is maximized if p is located on the x-axis given fixed p_0, p_1^+, and d.

Now consider the distance b from p_0 to p_k and the distance c from p_k to p_{k-1}^+. The robot travels at most $(2 - \frac{1}{2}\sin(2\alpha_k))b + c$ by Equation (1). Let q be the point where the line segment from p_k to p_{k-1}^+ intersects the y-axis. Let c_1 be the distance from p_k to q and c_2 the distance from q to p_{k-1}^+. Furthermore, let f be the distance from p_0 to q. We first show that $2f \geq b + c_1$. To see this we set $f = 1$. Then, $b + c_1 = (2 - \frac{1}{2}\sin(2\alpha_k))\cos\alpha_k + \sin\alpha_k$. The derivative of $b + c_1$ is $3\cos\alpha_k - 3\cos^3\alpha_k - 2\sin\alpha_k \leq 0$, for $0 \leq \alpha_k \leq \pi/2$ with the only real point of zero at $\alpha_k = 0$. Hence, 2 is the maximum of $b + c_1$ which proves the claim.

Finally, we consider the maximum of $f + c_2$ given p_{k-1}^+ where the line segment l

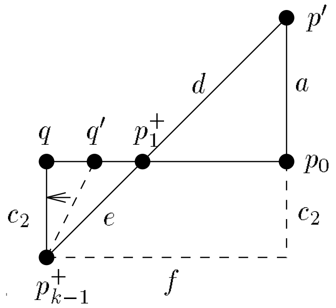

Figure 15: A triangle with a right angle can be formed.

from q to p_{k-1}^+ must form an angle of less or equal $270°$ with the y-axis. It is obtained if l is orthogonal to the y-axis. See moving q' to q in Figure 15. So assume q has the same y-coordinate as p_{k-1}^+. If we translate the line segment l such that q coincides with p_0 and the line segment from p_0 to q such that q coincides with p_{k-1}^+ (see Figure 15), then we obtain a triangle whose sides have length f, $c_2 + a$ and $d + e$ with a right angle in its lower left hand corner. The previous analysis applies and $\sqrt{5}(d + e) \geq 2f + c_2 + a \geq a + (2 - \frac{1}{2}\sin(2\alpha_k))b + c$ as claimed.

Theorem 2 *If P is a street with start point s and target point t, then there is a strategy for a robot with access to the local visibility map of its surroundings to travel from s to t on a path that is at most $\sqrt{5}$ times longer than the shortest possible route.*

5 Conclusions

We consider two problems in this paper. First we provide a lower bound of $\sqrt{2} - O(1/\sqrt{n})$ for the competitive ratio of any deterministic strategy that a robot may use to search in a rectilinear street if the coordinates of the target are given in advance to the robot. This implies that knowledge of the location of the target does not provide any advantage even for searching in rectilinear streets. We further show a similar result for $\mathcal{G}$-streets.

Secondly, we present a new strategy to search in arbitrarily oriented streets. We show a performance guarantee of $\sqrt{5}$ for our strategy though we believe that the analysis of our strategy is not tight and the actual competitive ratio may be significantly lower.

Unfortunately, the gap between the upper and lower bounds for searching in streets is still quite large and it seems that new ideas are needed to narrow it down.

A Analysis of ξ

To maximize ξ we can equivalently find the value of k that minimizes c/ξ, for c a constant. For ξ we obtain

$$\frac{2(\sqrt{2}-1)(m-k)}{\sqrt{2}\,[2m+(m-k)(m-k+1)]+2m-2k} = \frac{\sqrt{2}(\sqrt{2}-1)}{2m/(m-k)+m-k+1+\sqrt{2}} \Rightarrow$$

and, therefore,

$$\frac{c}{\xi} = \frac{2m}{m-k}+m-k+1+\sqrt{2}$$

Let

$$d(m,k) = \frac{2m}{m-k}+m-k+1+\sqrt{2},$$

then

$$\frac{\partial}{\partial k}d(m,k) = 2\frac{m}{(m-k)^2}-1$$
$$\frac{\partial^2}{(\partial k)^2}d(m,k) = 4\frac{m}{(m-k)^3}$$

With the critical points of $d(m,k)$ at $k_{1,2} = m \pm \sqrt{2m}$. Since $k \leq m$ we need only to study $k_1 = m - \sqrt{2m}$. Consider $d(m,\cdot)$ as a function of k. We see that the second derivative is positive for all $k < m$, and in particular at $k+1$. This implies that $d(m,\cdot)$ is minimized at k_1.

Thus for a fixed m, the best competitive ratio is attained at $k_1 = m - \sqrt{2m}$, namely $\sqrt{2}-\xi$ with $\xi = \sqrt{2}(\sqrt{2}-1)/(2\sqrt{2m}+1+\sqrt{2})$. Notice that ξ goes to zero as m goes to infinity.

References

[1] R. Baeza-Yates, J. Culberson, and G. Rawlins. "Searching in the plane", *Information and Computation*, Vol. **106**, (1993), pp. 234-252.

[2] A. Blum, P. Raghavan, and B. Schieber. "Navigating in unfamiliar geometric terrain", *Proc. of 23rd ACM Symp. on Theory of Computing*, (1991), pp. 494-504.

[3] K-F. Chan and T. W. Lam. "An online algorithm for navigating in an unknown environment", *International Journal of Computational Geometry & Applications*, Vol. **3**, (1993), pp. 227-244.

[4] A. Datta and Ch. Icking. "Competitive searching in a generalized street", *Proc. of 10th ACM Sypm. on Computational Geometry*, (1994), pp. 175-182.

[5] X. Deng, T. Kameda and C. Papadimitriou. "How to learn an unknown environment I : The rectilinear case", *Technical Report CS-93-04*, Department of Computer Science, York University, 1993. A preliminary

version appeared in *Proc. 32nd IEEE Symp. on Foundations of Computer Science*, (1991), pp. 298-303.

[6] Ch. Icking. Ph. D. Thesis, Fernuniversität Hagen, 1994.

[7] R. Klein. "Walking an unknown street with bounded detour", *Computational Geometry: Theory and Applications* **1**, (1992), pp. 325-351.

[8] J. Kleinberg. "On-line search in a simple polygon", *Proc. of 5th ACM-SIAM Symp. on Discrete Algorithms*, (1994), pp. 8-15.

[9] A. Mei and Y. Igarashi. "Efficient strategies for robot navigation in unknown environment" *Proc. of 21st International Colloquium on Automata, Languages and Programming*, (1994), to appear.

[10] C. H. Papadimitriou and M. Yannakakis. "Shortest paths without a map", *Theoretical Computer Science* **84**, (1991), pp. 127-150.

[11] D. D. Sleator and R. E. Tarjan. "Amortized efficiency of list update and paging rules", *Communications of the ACM* **28**, (1985), pp. 202-208.