Generalized Streets Revisited*

Alejandro López-Ortiz Sven Schuierer

Abstract

We consider the problem of a robot inside an unknown polygon that has to find a path from a starting point s to a target point t. We assume that it is equipped with an on-board vision system through which it can get the visibility map of its surroundings.

Furthermore, we assume that the robot is contained in a simple polygon that belongs to the class of generalized streets. We consider three problems.

1. We present a strategy that allows the robot to search for t in an arbitrarily oriented generalized street where the distance travelled by the robot under our strategy is at most 80 times the length of the shortest path from s to t.
2. We show that there are orthogonal generalized streets for which the distance travelled by the robot under any searching strategy is at least 9.06 times the length of the shortest path from s to t.
3. Finally, we show that even if the location of the target is known, there are orthogonal generalized streets for which the distance travelled by the robot under any searching strategy is at least 9 times the length of the shortest path from s to t.

1 Introduction

The problem of a robot searching for a target in an unknown environment has recently received a considerable amount of attention [2, 3, 4, 5, 8, 9, 11, 12, 13]. In this setting it is assumed that the robot is equipped with an on-board vision system that allows it to see its local environment. However, the robot does not have access to a complete map of its surroundings.

Since the robot has to make decisions about the search based only on the part of its environment that it has seen before, the search of the robot can be viewed as an *on-line* problem. One way to judge the performance of an on-line search strategy is to compare the distance traveled by the robot to the length of the shortest path from s to t. In other words, the robot's path is compared with that of an adversary's who knows the complete environment; this approach to analysing on-line algorithms was introduced by Sleator and Tarjan [14]. The ratio of the distance traveled by the robot to the optimal distance from s to t is called the *competitive ratio* of the search strategy.

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

Since, in general, the ratio between the distance a robot traverses and the length of a shortest path can be forced to be $\Omega(n)$ if the obstacles in the scene have a total of n edges, efforts have focussed on restricted classes of environments that allow more efficient search strategies. If, for instance, all the obstacles encountered by the robot are convex, then a competitive ratio of $Omega(\sqrt{n})$ is the best possible and can be achieved if the aspect ratio of the obstacles is bounded [7].

A different approach is to consider only one polygon which is now allowed to be non-convex. Klein was the first to consider a class of polygons called *streets* which allow search strategies that have a competitive ratio bounded by a constant [8]. A polygon is a street if the starting point s and the target are located on the polygon boundary and the counterclockwise boundary chain L from s to t is weakly visible to the clockwise boundary chain R from s to t and vice versa. Streets can be searched with a low constant competitive ratio [8, 9, 11]. Unfortunately, streets are a often too restrictive a class of polygons in order to model real environments.

In the search for larger classes of polygons that admit search strategies with a constant competitive ratio Datta and Icking propose a class of polygons they call *generalized streets* or $\mathcal{G}$-streets [5]. Here every point on the boundary is visible from some horizontal chord that connects L with R. The class of $\mathcal{G}$-streets can be shown to properly contain the class of streets. An even larger class is given by the class of HV-streets where every point in P is visible from a horizontal or a vertical chord that connects L and R [4]. Though the definitions of $\mathcal{G}$-streets and HV-streets include arbitrarily oriented polygons, the strategies that are presented only apply to orthogonal polygons, i.e. polygons whose edges are parallel to the coordinate axes. Datta and Icking present an algorithm that achieves a competitive ratio of nine in the L_1-metric and $\sqrt{82}$ (~ 9.06) in the L_2-metric. While nine can be shown to be a lower bound for any strategy, it is an open problem whether there is a better strategy in the L_2-metric. Similarly, Datta *et al.* present a strategy with an optimal competitive ratio of 14.5 w.r.t. the L_1-metric for searching in orthogonal HV-streets [4].

In this paper we present the first strategy to search in arbitrarily oriented $\mathcal{G}$-streets. Searching in arbitrarily oriented $\mathcal{G}$-streets turns out to be much more difficult than searching in orthogonal $\mathcal{G}$-streets which admit a very simple search strategy [5]. This situation is similar to exploring a simple polygon where a simple and optimal strategy is known for orthogonal polygons but no strategy to explore arbitrarily oriented polygons has been found yet [6]. We provide a strategy with a competitive ratio of 80 to search in $\mathcal{G}$-streets. Additionally we show that $\sqrt{82}$ is a lower bound for searching in orthogonal $\mathcal{G}$-streets, thus, proving that the strategy of Datta and Icking is optimal even in the L_2-metric. Finally, we also consider the problem of searching in an orthogonal $\mathcal{G}$-street where the location of the target is known. We show that this knowledge does not provide a significant advantage to the robot as there is also a lower bound of nine w.r.t. L_1-metric as in the case with unknown location of the target. This result was already mentioned in [11] but remained unproven.

The paper is organized as follows. In the next section we give some definitions. In Section 3 we present a strategy to search in arbitrarily oriented $\mathcal{G}$-streets. The lower bounds for orthogonal $\mathcal{G}$-streets with known and unknown location of the target are presented in Section 4. Finally, in Section 5 we summarize our result and conclude with some open problems.

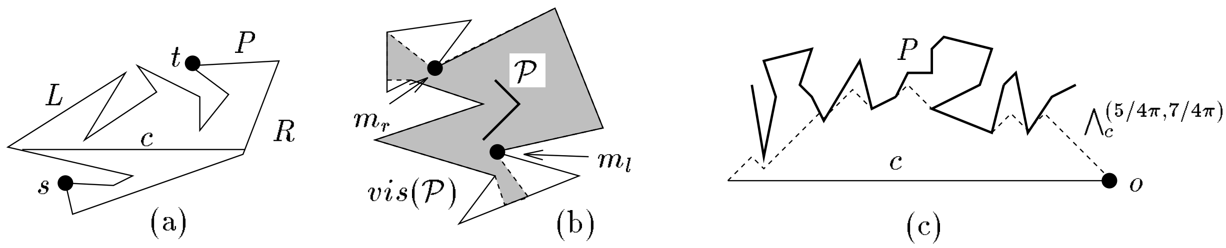

Figure 1: (a) A $\mathcal{G}$-street P with left chain L and right chain R. c is an LR-chord. (b) The visibility polygon $vis(\mathcal{P})$ of $\mathcal{P}$ with left $\mathcal{P}$-landmark m_l and right $\mathcal{P}$-landmark m_r. (c) The upper $(5/4\pi, 7/4\pi)$-envelope of c in P.

2 Definitions and Preliminary Results

We start out with some definitions concerning general geometric concepts. We view a *path or curve* as a continuous mapping from a closed interval into the two dimensional plane. In particular, we assume paths to be oriented from their start point to their end point. Let $\mathcal{C}$ be a simple, closed curve consisting of n line segments such that no two consecutive segments are collinear. We define a *simple polygon* P to be the union of $\mathcal{C}$ and its interior.

If p and q are two points, then we denote the line segment spanned by p and q by $\overline{pq}$. The (oriented) straight path from p to q is denoted by $\overrightarrow{pq}$.

A line segment c inside P is called a *chord* if the end points of c are on the boundary of P. A *maximal line segment* or a *maximal chord* l in P is a line segment inside P such that there is no other line segment that properly contains l and that is also contained in P.

Now we turn to generalized streets and their properties. If P is a polygon and s and t are two points on the boundary of P, then we call the counterclockwise polygonal chain from s to t the *left chain* and denote it by L; the clockwise polygonal chain from s to t is called the *right chain* and denoted by R. A chord c inside P is an *LR-chord* if c intersects both L and R.

Definition 2.1 ([5]) *Let P be a polygon and s and t two points on the boundary of P with left chain L and right chain R. P is called a* generalized street *or $\mathcal{G}$-street if every point in P is visible from a horizontal LR-chord (see Figure 1a).*

LR-chord are used to guide the search for t.

Lemma 2.1 ([5]) *c is an LR-chord if and only if a shortest path from s to t intersects c.*

We also need some definitions that deal with visibility.

Definition 2.2 *Let P be a polygon and $\mathcal{P}$ be a path in P. The set of points that are visible to $\mathcal{P}$ is called the* visibility polygon of $\mathcal{P}$ *and denoted by $vis(\mathcal{P})$ (see Figure 1b).*

A maximal line segment of the boundary of $vis(\mathcal{P})$ that does not belong to the boundary of P is called a *window* of $vis(\mathcal{P})$. A window w splits P into two parts one of which contains $vis(\mathcal{P})$. The part that does not contain $vis(P)$ is called the *pocket of* w. The end point of a window that is closer to $\mathcal{P}$ is called a $\mathcal{P}$*-landmark*. If m is a $\mathcal{P}$-landmark and w is the window it belongs to, then w is called the window *induced* by m; furthermore, if Q is the pocket of w, the Q is called the *pocket induced by* w *or by* m. We assume that a window is oriented such that the landmark that induces it is its start point. A *left* window is a window whose induced pocket is to the left w.r.t. its orientation. A landmark is called a *left* landmark if it induces a left window. A right window and a right landmark are defined analogously (see also Figure 1b).

3 Searching in a Generalized Street

In this section we describe how to search for a target in an arbitrarily oriented generalized street. The general idea is to advance from one horizontal LR-chord to the next until the target t is seen. This is the same general strategy as used by Datta and Icking to search orthogonal $\mathcal{G}$-streets [5]. The crucial step of the strategy is, of course, to identify a new LR-chord c' if we are given an LR-chord c. This is done by exploring the neighbourhood of the chord c. If we call the first point the robot reaches on c the *arrival point of* c, then we have to ensure that the amount traveled by the robot in the exploration is proportional to the distance from the arrival point of c to the arrival point of c'.

3.1 Searching the Neighbourhood of a Chord

In the following we assume that the robot is located at the arrival point o on a horizontal LR-chord c. The robot uses four paths to explore the neighbourhood of c, one for each quadrant of the coordinate system with the origin in the arrival point at c and the x-axis collinear with c.

Before describing our strategy in detail, we briefly discuss a method due to Baeza-Yates *et al.* [1] for searching a point in m concurrent rays. We assume that the robot is placed at the origin of m concurrent rays and it has to find a point t which is situated in one of the rays. The distance of the point t is unknown to the robot though it knows a lower bound ε; the robot can only detect the point t when it reaches t. In the strategy of Baeza-Yates *et al.* the robot visits the rays one by one in a round robin fashion until the point t is reached. In every ray, the robot goes a certain distance and if t is not reached, returns and explores the next ray. The distance from the origin the robot travels before the i-th turn is given by $(m/(m-1))^{i-1}\varepsilon$. The competitive ratio of this strategy is $1 + 2m^m/(m-1)^{m-1}$ which can be shown to be optimal [1].

For two concurrent rays (i.e., $m = 2$), the robot executes cycles of steps in increasing powers of 2 and the competitive ratio is 9. Similarly, for four concurrent rays, the robot executes cycles of steps in increasing powers of $\frac{4}{3}$ and the competitive ratio is 19.98.

We now turn to describing the shape of the paths that the robot explores in detail. We say the robot θ*-sees* the line segment c if the ray originating from the robot at an angle of θ intersects c before it intersects any points of the exterior of P.

Definition 3.1 *Let c be a chord in P and $\pi < \sigma < \tau < 2\pi$. The* upper (σ, τ)-envelope *of c is the set of the highest points above c which σ- and τ-see c. We denote the upper (σ, τ)-envelope of c by $\bigwedge_c^{(\sigma,\tau)}$. If $0 < \sigma < \tau < \pi$, then the* lower (σ, τ)-envelope *of c is defined analogously and denoted by $\bigvee_c^{(\sigma,\tau)}$ (see Figure 1c).*

With this definition we now can give a precise description of how the robot searches the neighbourhood of c.

Algorithm Landmark Detection
Input: a $\mathcal{G}$-street P, a horizontal LR-chord c in P, and the arrival point o of the robot on c;
Output: a c-landmark m;
let c_l be the part of c to the left of o and c_r be the part of c to the right of o;
let d be the radius of largest ball around p that is contained in P;
$dir := l$; (* The algorithm explores the neighbourhoods of c_l and c_r alternatingly *)
while no c-landmark has been detected **do**
  trace the upper $(5/4\pi, 7/4\pi)$-envelope of c_{dir} for a distance of at most d;
  return to o; $d := 4/3 * d$;
  trace the lower $(\pi/4, 3/4\pi)$-envelope of c_{dir} for a distance of at most d;
  return to o; $d := 4/3 * d$;
  change the direction dir;

We still have to specify how the robot identifies a c-landmark. Note that a reflex vertex v that induces a window from the current robot position is not necessarily a c-landmark as v may see points of c that the robot has not reached yet.

3.2 Direct Detection of a Landmark

In the following we only consider the detection of c-landmarks above c by traveling on $\bigwedge_{c_l}^{(\sigma,\tau)}$ and $\bigwedge_{c_r}^{(\sigma,\tau)}$. The detection of c-landmarks below c is, of course, analogous. There are several ways for the robot to detect a c-landmark. The first way is to look for c-landmarks which induce *bottom windows.*

Definition 3.2 *If $\mathcal{P}$ is a curve in P, then the* horizontally visible polygon *induced by $\mathcal{P}$ is the set of points in P that can be reached by a horizontal ray from a point on $\mathcal{P}$.*

If Q is the horizontally visible polygon induced by $\mathcal{P}$, then the line segments of the boundary of Q that are not part of the boundary of P are again called the *windows* of Q. Clearly, all the windows of Q are horizontal. A window w is a *top window* if there are points of $P \setminus Q$ above w and a *bottom window*, otherwise.

Lemma 3.1 *Let Q be the horizontally visible polygon induced by a path $\mathcal{P}$ that starts on c. If w is a bottom window of Q, then the maximal horizontal line segment containing w is an LR-chord.*

Lemma 3.1 implies that as soon as the robot detects a bottom window it has found a new LR-chord and it stops the search for a new LR-chord. Hence, we assume in the following that there are only top windows in the horizontally visible polygon induced by the path of the robot.

A second possibility to identify a c-landmark m is to cross the maximal line segment l containing the window induced by m. Note that if w is a window, then l intersects a reflex vertex v such that v is between c and m and both edges incident to v are on the opposite side of l as the edges incident to m.

Lemma 3.2 *Let c be a horizontal LR-chord, m a c-landmark that induces window w, and l the maximal line segment that contains w. If v is a reflex vertex intersected by l between c and m such that the edges incident to v are on the opposite side of l as the edges incident to m, then the maximal horizontal line segment that contains v is an LR-chord.*

Note that if v is between the intersection point p of the path of the robot with l and the c-landmark m, then v is above p. In this case the robot moves to the maximal horizontal line segment through v and stops the search for a new LR-chord.

Finally, there is third case in which the robot can immediately detect a c-landmark. Recall that a left c-landmark is a c-landmark which induces a window w whose pocket is locally to the left of w.

Lemma 3.3 *If p_l is on $\bigwedge_{c_l}^{(\sigma,\tau)}$ and the left c-landmark m first becomes visible to the robot at p_l, then there is a new LR-chord whose distance to the origin o of $\bigwedge_c^{(\sigma,\tau)}$ is at least as large as the distance from o to p_l.*

Note that in all three of the above case we have identified a new LR-chord whose L_1-distance is at least as large as the vertical distance between the current robot position and c.

3.3 Indirect Detection of a Landmark

Unfortunately, the above possibilities to discover c-landmarks are not exhaustive. In the following we show that there are certain regions associated with the robot position which are completely visible to the robot if they do not contain a c-landmark. We denote the ray starting in point p with orientation θ by $r_\theta(p)$.

Definition 3.3 *Let c be a chord in P and $\pi < \sigma < \tau < 2\pi$. If p is a point in $\bigwedge_c^{(\sigma,\tau)}$ and q is a point on $r_\sigma(p)$ ($r_\tau(p)$), then q is a* critical point *if q is an end point of c or q is between p and c and a point of the boundary of P.*

Definition 3.4 *Let c be a horizontal line segment in P, $\pi < \sigma < \tau < 2\pi$, and p a point in $\bigwedge_c^{(\sigma,\tau)}$.*

(i) *If $r_\sigma(p)$ intersects a critical point, then p is called a* left extreme point *of $\bigwedge_c^{(\sigma,\tau)}$.*

(ii) If $r_\tau(p)$ intersects critical point, then p is called a right extreme point *of $\bigwedge_c^{(\sigma,\tau)}$.*

(iii) If p is left and right extreme point of $\bigwedge_c^{(\sigma,\tau)}$, then p is called a peak *of $\bigwedge_c^{(\sigma,\tau)}$.*

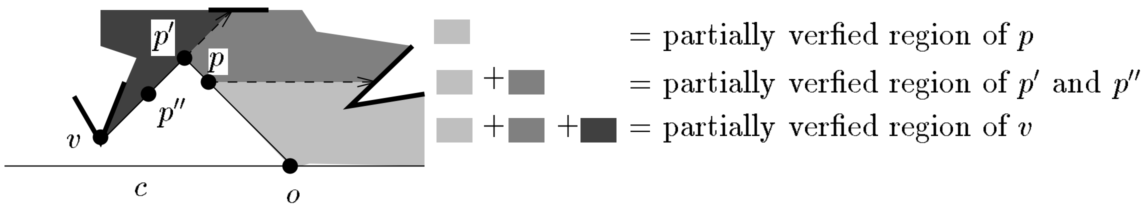

Figure 2: The upper $(5/4\pi, 7/4\pi)$-envelope of c contains the right extreme point p, the peak p', the left extreme point p'', and the valley v.

(iv) If p is neither a left nor a right extreme point of $\bigwedge_c^{(\sigma,\tau)}$, then p is called a valley *of $\bigwedge_c^{(\sigma,\tau)}$.*

The definition is illustrated in Figure 2. Note that if the point p of $\bigwedge_c^{(\sigma,\tau)}$ is not part of the boundary of P, then either $r_\sigma(p)$ or $r_\tau(p)$ intersects the boundary of P. The peaks of $\bigwedge_c^{(\sigma,\tau)}$ are the local maxima of $\bigwedge_c^{(\sigma,\tau)}$ and a point p is a valley of $\bigwedge_c^{(\sigma,\tau)}$ if all neighbourhoods of p intersect the exterior of P above c.

If p is a point in P and $\sigma \in [0, 2\pi)$, then we denote the maximal line segment that is part of the beginning of $r_\sigma(p)$ and that is contained in P by $\overrightarrow{p\sigma}$.

Definition 3.5 *Let p_l be a point of $\bigwedge_{c_l}^{(\sigma,\tau)}$ and $\mathcal{P}$ the path consisting of the concatenation of c_r with the part of $\bigwedge_{c_l}^{(\sigma,\tau)}$ from o to p. The* partially verified region of p *is defined as follows (see Figure 2).*

1. *If p is a right extreme point of $\bigwedge_c^{(\sigma,\tau)}$, then the* partially verified region of p_l *is the simple polygon to the right of the path consisting of $\overrightarrow{p_l 0}$ concatenated with $\mathcal{P}$.*
2. *If p_l is a peak of $\bigwedge_c^{(\sigma,\tau)}$, then the* partially verified region of p_l *is the simple polygon to the right of the path consisting of $\overrightarrow{p_l(\sigma - \pi)}$ concatenated with $\mathcal{P}$.*
3. *If p_l is a left extreme point of $\bigwedge_c^{(\sigma,\tau)}$ and q is the last peak before p_l, then the* partially verified region of p_l *is the partially verified region of q.*
4. *If p_l is a valley of $\bigwedge_c^{(\sigma,\tau)}$, then the* partially verified region of p_l *is the simple polygon to the right of $\mathcal{P}$.*

If p_r is a point on $\bigwedge_{c_r}^{(\sigma,\tau)}$, then the partially verified region is defined by considering the mirror image of P along the vertical line through o. The partially verified region of a point p_l on $\bigwedge_{c_l}^{(\sigma,\tau)}$ is important since all the points in this region that can be seen by c_l can also be seen by the robot.

Lemma 3.4 *If p_l is a point in $\bigwedge_{c_l}^{(\sigma,\tau)}$, then all the points in the partially verified region R_l of p_l that are visible to c_l are also visible to the part of $\bigwedge_{c_l}^{(\sigma,\tau)}$ from o to p_l. A similar statement holds for the points p_r in $\bigwedge_{c_r}^{(\sigma,\tau)}$.*

If we intersect the partially verified region of the robot position p_l with the partially region of the last robot position p_r on $\bigwedge_{c_r}^{(\sigma,\tau)}$, then we obtain a new region which is completely visible to the robot.

Definition 3.6 *If p_l is a point of $\bigwedge_{c_l}^{(\sigma,\tau)}$ and p_r is point of $\bigwedge_{c_r}^{(\sigma,\tau)}$, then the* verified region of p_l and p_r *is the intersection of the partially verified region of p_l and the partially verified region of p_r.*

The verified region now enables the robot to detect the remaining c-landmarks. If there are no c-landmarks, then the whole verified region is visible to the robot.

Lemma 3.5 *Let p_l be a point on $\bigwedge_{c_l}^{(\sigma,\tau)}$ and p_r a point on $\bigwedge_{c_r}^{(\sigma,\tau)}$. If there is no c-landmark in the verified region R of p_l and p_r, then all the points in R are seen by union of the part of $\bigwedge_{c_l}^{(\sigma,\tau)}$ from p_l to o and the part of $\bigwedge_{c_r}^{(\sigma,\tau)}$ from p_r to o.*

Lemma 3.5 implies in particular that the verified region of the end points c_l and c_r contains the upper visibility polygon of c since the end points of $\bigwedge_{c_l}^{(\sigma,\tau)}$ and $\bigwedge_{c_r}^{(\sigma,\tau)}$ are the end points of c which are both valleys.

3.4 Eliminating Landmarks

After having identified a c-landmark we are able to decide whether t is above or below c. In the following we assume that we have discovered a c-landmark m above c. Hence, the robot can stop the search paths below c. If a c-landmark has been discovered by one of the cases described by the Lemmas 3.1, 3.2, or 3.3, then there is a new LR-chord c' whose L_1-distance to o is at least as large as the vertical distance of the current robot position p to c. The robot computes the closest point o' of c' to o, moves to it, and starts exploring the neighbourhood of c' by Algorithm *Landmark Detection* again.

If a c-landmark is detected by Lemma 3.5, then this also implies that there is an LR-chord c' that is above c. However, the distance of c' to c may be so small that the advancement from c to c' does not pay for the searching effort. Moreover, there may be many c-landmarks and at this point the robot is not able to decide behind which c-landmark the shortest path from s to t continues. Hence, we use a 2-way ray search to identify a c-landmark which induces an LR-chord or to eliminate pockets of c-landmarks as possible locations of t. In order to do this we extend c to the left and the right by the maximal y-monotone boundary chains of P that start at the left resp. the right end point of c. Hence, $\bigwedge_{c_l}^{(\sigma,\tau)}$ now denotes the local upper (σ,τ)-envelope of the concatenation of c_l and the maximal upper y-monotone boundary chain $\mathcal{C}_l$ starting at the left end point of c_l. If the path of the robot ends at a point of $\mathcal{C}_l$, then the robot continues upwards following $\mathcal{C}_l$ and traveling along a $3/4\pi$-oriented ray where ever possible. $\bigwedge_{c_r}^{(\sigma,\tau)}$ is defined similarly. The concatenation of $\bigwedge_{c_l}^{(\sigma,\tau)}$ with $\bigwedge_{c_r}^{(\sigma,\tau)}$ is called the *extended upper envelope of* c and denoted by $\mathcal{P}$. $\bigwedge_{c_l}^{(\sigma,\tau)}$ is called the *left arm of* $\mathcal{P}$ and $\bigwedge_{c_r}^{(\sigma,\tau)}$ the *right arm of* $\mathcal{P}$. If c' is an LR-chord above c, then we denote the extended upper envelope of c' by $\mathcal{P}'$.

Let V_{p_l} be the intersection of the partially verified region of p_l with the visibility polygon of $\mathcal{P}$. We distinguish five cases how a right c-landmark can be added to V_{p_l} depending on the location of the robot. In all cases we maintain the following invariant.

> *If p_l is a left extreme point on the left arm of $\mathcal{P}$, then there is no right c-landmark in V_{p_l}.*

Note that the robot needs only be concerned with right c-landmarks by Lemma 3.3. We assume that p_l is the current robot position.

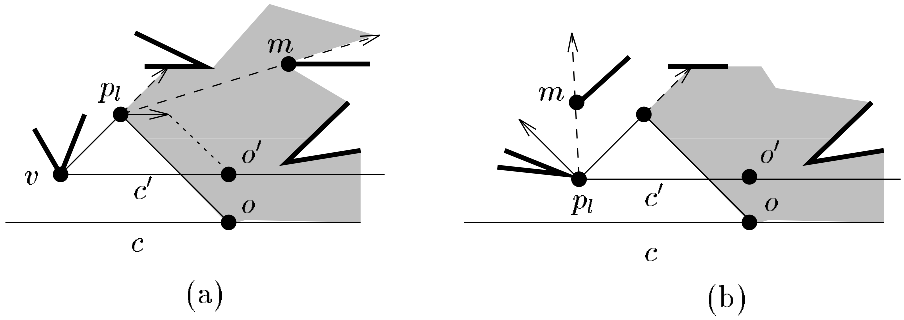

Figure 3: (a) An illustration of Case 2. The robot moves from o to the peak p_l where it detects a c-landmark m. It then moves horizontally to the right until it intersects $\mathcal{P}'$. (b) An illustration of Case 4. The robot moves from o to the valley p_l where it detects a c-landmark m. c' is the new search chord and o' the new origin. However, the robot continues on the $3/4\pi$-ray starting in p_l.

Case 1 The point p_l is a right extreme point and right a c-landmark m is added to V_{p_l}. Since m is added to V_{p_l} and p_l is a right extreme point, m has the same height as p. Furthermore, m induces a right window w. Since p_l is collinear with w and w is a right window, w is a bottom-window of the horizontally visible polygon of $\mathcal{P}$. By Lemma 3.1 it induces an LR-chord c'. The robot moves to c' and stops the search.

Case 2 The point p_l is a peak and a right c-landmark m is added to V_{p_l}.
There is a reflex vertex v that intersects the ray r with orientation $5/4\pi$ between p_l and the extension of c. It is easy to see that there is a point in pocket of the c-landmark m that cannot see any point below the right chord through v. By Lemma 3.2 the maximal horizontal line segment c' through v is an LR-chord. Hence, c' is now the new search chord.

The robot moves horizontally to the right until it intersects $\mathcal{P}'$. Note that the distance traveled by the robot to intersect $\mathcal{P}'$ is at most the distance between c and c'. The new position of the robot is not a peak anymore—even if the robot has not moved—since the vertex v is now part of the extension of c' and the invariant still holds. Furthermore, the distance that the robot follows the left and right arms of the extended upper envelope of c' is now reduced by the distance from o to c. For illustration refer to Figure 3a.

Case 3 The point p_l is a left extreme point and a right c-landmark m is added to V_{p_l}. Since the verified region of a left extreme point is the same as the verified region of its peak, this case does not occur.

Case 4 The point p_l is a valley and a right c-landmark m is added to V_{p_l}.
Note that in this case there are no right c-landmarks in the part of V_{p_l} that is to the right of the maximal $5/4\pi$-oriented line segment through p_l. If c' is the maximal horizontal line segment through p_l, then c' becomes the new search chord and the new origin o' is the closest point of c' to o. However, the robot does not return to o'; instead, it continues its search on the left arm of the extended upper $(5/4\pi, 7/4\pi)$-envelope of c' where the

origin is now considered to be p_l until the robot has traversed the distance d of Algorithm *Landmark Detection*. Note that the invariant still holds in this case; see Figure 3b.

Only then does it return to o' and continues the search where as in Case 2 the distance that the robot follows the left and right arms of the extended upper envelope of c' is reduced by the distance from o to o'.

Case 5 The robot has reached the left boundary and a right c-landmark m is added to V_{p_l}.
The partially verified region of the robot position is defined as the partially verified region of a right extreme point. Therefore, the case can be handled analogous to Case 1.

Apart from the above cases the robot, of course, also considers the cases according to the Lemmas 3.1, 3.2, and 3.3.

At the end of the search we have either discovered a new LR-chord above the current robot position or eliminated all but one c-landmark as the following lemma shows.

Lemma 3.6 *If the left arm stops, i.e. there robot has followed the whole length of the left arm, then there are no right c-landmarks.*

3.5 Analysis of the Strategy Landmark-Searching

If we use the optimal strategy to explore four concurrent rays, then the robot travels at most 19.98 farther than the length of the last explored arm of one of the extended envelopes in order to identify the next LR-chord.

It can be easily seen that the following invariant holds during the search: If p_l is a point on the left arm of $\mathcal{P}$, then the L_1-distance of all c-landmarks to the arrival point on c is at least α times the length of the current explored arm of $\mathcal{P}$ from o to p_l where

(i) $\alpha = 1/\sqrt{2}$ if p_l is a valley.

(ii) $\alpha = 1/\sqrt{2}$ if p_l is a left extreme point or a peak and Case 3 has not occurred; (note that we do not have to consider right extreme points as the verified region of a right extreme point is the same as the verified region of the preceding peak;)

(iii) $\alpha = \sqrt{2}/3$ if p_l is a left extreme point or a peak and Case 3 has occurred;

(iv) $\alpha = 1/(\sqrt{2}+1)$ if the robot has reached the left boundary and Case 3 has not occurred and

(v) $\alpha = 1/(2\sqrt{2})$ if the robot has reached the left boundary and Case 3 has occurred;

Since the robot travels at most 20 times the distance of an explored arm and at the end the robot either travels to an LR-chord above the current robot position or to a c-landmark, the total distance traversed by the robot from one arrival point to next is at most $2\sqrt{2}\cdot 20$ the L_1-distance between them. Finally, we need to combine these local estimates to obtain a bound on the complete path traveled by the robot.

Lemma 3.7 *If $o_1, o_2, \ldots, o_k$ is the sequence of arrival points at the LR-chords visited by the robot, then there is an L_1-shortest path from s to t that contains all o_i, $1 \le i \le k$, in this sequence.*

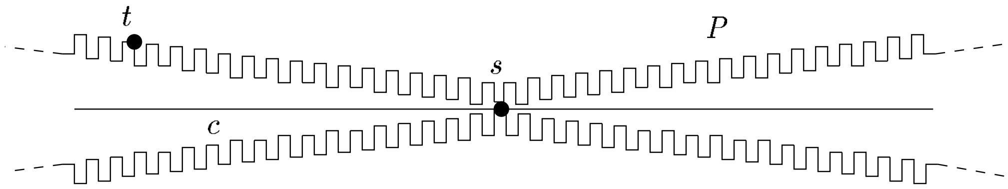

Figure 4: A $\mathcal{G}$-street which forces a competitive ratio of $\sqrt{82}$.

With the help of Lemma 3.7 it is easy to prove the main result of this section.

Theorem 1 *The total distance traveled by a robot using the Algorithm Landmark Detection is at most* $40\sqrt{2}$ *times the* L_1*-distance and at most* 80 *times the* L_2*-distance between* s *and* t.

4 Lower Bounds

4.1 A Lower Bound for the L_2-Distance

We first present the lower bound for searching in orthogonal $\mathcal{G}$-streets if the distance is measured in the L_2-metric and the location of the target is unknown. Consider the $\mathcal{G}$-street P in Figure 4. The target t can be hidden in any of the teeth of P and P still is a $\mathcal{G}$-street. In order to decide whether the target t is contained in a tooth T, the robot must intersect the vertical line through the rightmost point of T if T is to the left of s and the the vertical line through the leftmost point of T if T is to the right of s. If P contains enough teeth, then the robot is forced to travel at least nine times the horizontal distance of s to the tooth that contains t [1]. It does not pay for the robot to leave the chord c since if the path of the robot goes above c, then an adversary places t in a tooth below c and vice versa. If p is the point on c at which the robot sees t, then the robot travels a distance of $9d(s,p) + d(p,t)$ while the L_2-shortest path has length $\sqrt{d(s,p)^2 + d(p,t)^2}$. By choosing $d(p,t) = 1/9d(s,p)$, i.e., by putting the teeth of P along lines with slopes $1/9$ and $-1/9$, respectively, we obtain a competitive ratio of $\sqrt{82}$ as claimed.

Now consider the situation in which the robot searches for a target of known location on a $\mathcal{G}$-street of unknown shape. In this case the polyon of Figure 4 no longer provides a 9 lower bound. We show, however, that searches in this case are still $\Omega(9)$ competitive. For this we use the following result shown in [10] which applies to the case of of a robot searching for a target on the real line using an asymmetric strategy.

Theorem 2 *Let* C_S^L (C_S^R) *be the competitive ratio for finding a target point on a real line on the left (right) under a given strategy* S*. Then* $(C_S^L + C_S^R)/2 \geq 9$.

With this theorem we can now prove the main result.

Theorem 3 *Searching for a target of known location on a rectilinear* $\mathcal{G}$*-street is at least* 9*-competitive.*

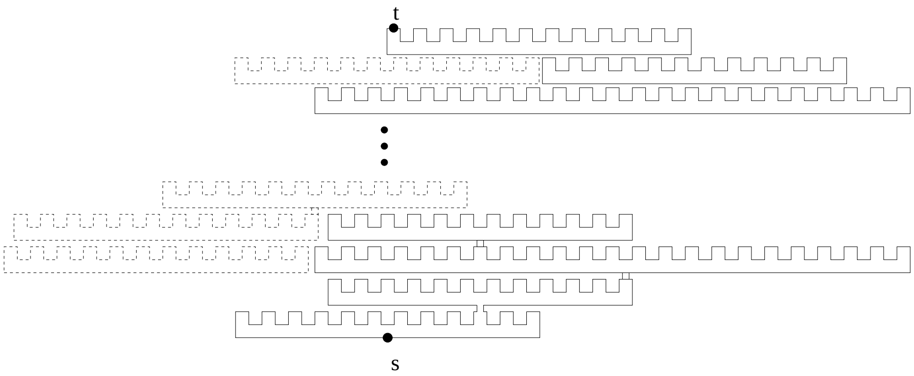

Figure 5: A Lego stack polygon.

Proof: Let n be the distance between s and t. Without loss of generality let the origin be the initial position of the robot and $(0, n)$ the position of the target. Figure 5 shows the polygon (solid lines). The dashed polygons represent other alternative polygons which can also result in polygons within this class. Each polygon in the family is made of M connected rake polygons between s and t. A connection point joins a tooth from the bottom rake to the middle of the top rake.

Rakes are numbered in the order of occurrence on the robot's path from s to t. Each rake i has height n/M or $1/(nM)$; it is symmetrically centered above the entrance point, and has length $2n$. Initially the robot sees only one rake, and searches each tooth for the opening to the next rake. (The robot knows that the target is not within a tooth, as the coordinates of the target are known to the robot).

As is usual with adversarial arguments, the adversary constructs the polygon on-line depending on the robot's moves. Let r_i be the x-coordinate of the entrance point to the ith polygon (assume that $r_i > 0$). Let D_i be the distance from the entrance point to the exit (connecting) tooth. Let C_i be the ratio of the traversed by the robot in rake i divided over D_i. A sequence $\{w_i\}$ is ϵ-increasing if $w_{i+1} \geq w_i + \epsilon$.

Adversary's Strategy

> The adversary selects a target number of rakes M that make the polygon. The height of each rake is thus, in principle, n/M. The adversary aims to create a polygon with a total optimal distance of at least $nM/2$. This gives an average of $n/2$ units per rake. To achieve this desired optimal path length, the adversary determines the height of each rake as follows:
>
> If on a given rake, the robot forces a shorter optimum path shorter than $n/2$, the adversary makes the n subsequent rakes of height $1/M$ each (see figure 6). Since on each rake the optimum path is at least a unit long, the optimum path is at least n units longer when it reaches the next regular height rake, for an average gain of $N/2$ per each $1/M$ height gain.

- Let $i \leftarrow 1$. Without loss of generality, the adversary opens a tooth on the right side, with competitive ratio $C_1 \geq 9 - \epsilon$. Let $b \leftarrow 1$; $\quad \mathcal{R}[b] \leftarrow (C_1, D_1)$.
- For each i from 2 to M do

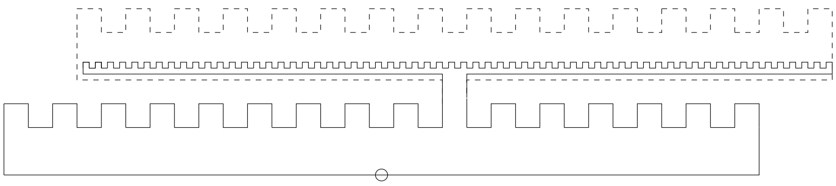

Figure 6: Variable height rakes.

Case 1: If the robot reaches a tooth in $[0, r_i]$ with competitive ratio C_i such that $\frac{1}{2}(C_i + \mathcal{R}[b,1]) \geq 9 - \epsilon/2$, then the adversary opens that tooth.

- If $\mathcal{R}[b,2] - D_i < n/2$ then
  Let $\mathcal{R}[b-1,2] \leftarrow \mathcal{R}[b,2] - D_i + \mathcal{R}[b-1,2]$.
  Let $\mathcal{R}[b-1,1] \leftarrow (\mathcal{R}[b,1](\mathcal{R}[b,2] - D_i) + \mathcal{R}[b-1,1]\mathcal{R}[b-1,2])/\mathcal{R}[b-1,2]$.
  Let $b \leftarrow b-1$.
- Else let $\mathcal{R}[b,2] \leftarrow \mathcal{R}[b,2] - D_i$.

Case 2: Else let $b \leftarrow b+1$; $\quad \mathcal{R}[b] \leftarrow (C_i, D_i)$.
Invariant: the sequence of competitive ratios $\mathcal{R}[b,1]$ is ϵ-increasing and $\mathcal{R}[b,2] \geq n/2$.
The adversary opens an alley to the right of the entrance point at a competitive ratio $C_i = C_i^R$ such that $\frac{1}{2}(C_i^L + C_i^R) \geq 9$.

- In the Mth polygon, the robot knows that its present position is horizontally aligned with the target and moves directly towards it. In this case, the adversary does not oppose the robot's move, and e the robot reaches the target optimally within R_M.

For case 2, first note that, if the invariant holds, then theorem 2 implies that it is always possible to choose an entrance point as requested. To prove the invariant we note that if we are in case 2, then the worst-case competitive ratio for all points on the left C_i^L is such that $\frac{1}{2}(C_i^L + R[b,1]) \leq 9 - \epsilon/2$ which implies $R[b,1] \leq 18 - C_i^L - \epsilon$. But we know from theorem 2 that $\frac{1}{2}(C_i^L + C_i^R) \geq 9$. Thus $C_i^R \geq 18 - C_i^L$ which implies $C_i^R \geq 18 - C_i^L \geq R[b,1] + \epsilon$.

Thus case 2 ensures that, if the exit alley is to the right, the competitive ratio increased at least by ϵ, while Case 1 ensures that if the alley is on the left, the robot traverses at least $n/2$ units which **together** with a previous right move balance out to an over 9-competitive ratio. In this case, the step is eliminated from the sequence of right moves as it has been "cancelled out" by the left move. Let $\epsilon = 1/n^2$ and $M = n^4$.

It follows that if the the robot follows a strategy which has only case 2 adversarial moves, the robot reaches the last polygon having traversed at least $(9 + M/n^2)/2 \times n^4$, and it is n^4 units away from the target, for a total competitive ratio of $(9 + n^2)/2 + 1$ which is arbitrarily large. Therefore the robot must choose a number of case 1. If all of moves are case 1, once again we obtain a trivial lower bound of 9 for the competitive ratio. As we shall see, the total distance traversed by the robot is at least

$$b \cdot \frac{n}{2}\frac{(9 + b\epsilon)}{2} + b\frac{n}{2} + 9n(M - b) + 5\frac{n}{2}\frac{b}{2}.$$

The first term denotes the fact that in each of the b case 2 configurations the robot traversed at least $n/2$ units. The competitive ratio, for the first $n/2$ units is the average of all competitive ratios in $\mathcal{R}$ which comes to $(9 + b\epsilon)/2$. As the movement in the first term was to the right, the second term denotes the optimal trajectory back to the target. The third term expresses the fact that in the remaining other $M-b$ cases, the competitive ratio was at least 9, and the total distance traversed was at least n ($n/2$ units to the right and $n/2$ units back to the left). The last term accounts for the fact that the robot may traverse between $n/2$ and n units at any competitive ratio. Thus the robot may wish to maximize the distance traversed at "low" competitive ratios which occur at the beginning. The lowest competitive ratio to the right is 9, and each distance must be traversed to and fro, for a total competitive ratio of $(9+1)/2 = 5$. Such low competitive ratio can be attained in at most half of the case 2 situations.

The optimal distance is given then by $bn+n(M-b)+n\frac{b}{2}$. By differentiating we see that the competitive ratio is maximized when either $k = 0$ or $k = -2n^4 + 2\sqrt{n^8 + 19n^6}$. In the first case is easy to see that the competitive ratio is 9. For the second case, substituting we obtain

$$2n\sqrt{n^2+19} - 2n^2 - 10 = \frac{2n(\sqrt{n^2+19} - n)(\sqrt{n^2+19} + n)}{\sqrt{n^2+19} + n} - 10 = 9$$

as required. □

5 Conclusions

We have presented a strategy to search in arbitrarily oriented generalized streets. It uses a new approach to search the neighbourhood of a chord which efficiently identifies and eliminates c-landmarks if the current search chord is c. The competitive ratio of our search strategy is bounded by 80. Furthermore, we have presented two lower bounds. One, a simple example, settles the competitive ratio of searching in orthogonal $\mathcal{G}$-streets w.r.t. the L_2-metric. We show that $\sqrt{82}$ is also a lower bound. Secondly, we investigate if it is an advantage for the robot if it is given the location of the target in advance. We show that there are polygons for every strategy that force the robot to walk at least nine times the length of the shortest path from s to t.

An important open problem is the competitive ratio of searching in $\mathcal{G}$-streets. We have provided an upper bound. The best lower bound known is $\sqrt{82}$ which also applies to orthogonal $\mathcal{G}$-streets. Though our algorithm is most probably not optimal, we suspect that a much higher lower bound than nine can be shown. The question also remains if there are still larger classes of arbitrarily oriented polygons that can be searched at a constant competitive ratio. HV-streets seem to be a natural candidate. However, it seems that new and simpler ideas have to be developed in order to obtain a strategy to search arbitrarily oriented HV-streets.

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 on-line algorithm for navigating in an unknown environment", *International Journal of Computational Geometry & Applications*, Vol. **3**, (1993), pp. 227-244.

[4] A. Datta, Ch. Hipke, and S. Schuierer. "Competitive searching in polygons—beyond generalized streets". In *Proc. Sixth Annual International Symposium on Algorithms and Computation*, pages 32–41. LNCS 1004, 1995.

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

[6] 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.

[7] B. Kalyasundaram and K. Pruhs. "A competitive analysis of algorithms for searching unknown scenes", *Computational Geometry: Theory and Applications* **3**, (1993), pp. 139-155.

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

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

[10] A. López-Ortiz. "On-line Searching on Bounded and Unbounded Domains". *Ph.D. Thesis*, Department of Computer Science, University of Waterloo, (1996).

[11] A. López-Ortiz and S. Schuierer. "Going home through an unknown street". In S. G. Akl, F. Dehne, and J.-R. Sack, editors, *Proc. 4th Workshop on Algorithms and Datastructures*, pp. 135–146. LNCS 955, (1995).

[12] 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.

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

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