Searching and On-line Recognition of Star-Shaped Polygons

Alejandro López-Ortiz *,

School of Computer Science, University of Waterloo,
Waterloo, Ontario, Canada, N2L 3G1

Sven Schuierer [b]

[b]*Institut für Informatik, Universität Freiburg,*
Am Flughafen 17, Geb. 051, D-79110 Freiburg, FRG

Abstract

We study the problem of on-line searching for a target inside a polygon. In particular, we propose a strategy for finding a target of unknown location in a star-shaped polygon with a competitive ratio of 11.52. We also provide a lower bound of 9 for the competitive ratio of searching in a star-shaped polygon which is close to the upper bound.

A similar task is the on-line recognition of a star-shaped polygon P. Here, the robot travels on a path that allows it to decide whether P is star-shaped or not. We present a strategy with a competitive ratio of 28.85 and give a lower bound of $\sqrt{82}$ for this problem.

Key words: On-line searching, algorithmic foundations of robotics, competitive ratio, motion planning, star-shaped polygons.

1 Introduction

In the past years on-line searching has been an active area of research in Computer Science (e.g. [1–3,12,14]). In its full generality, an on-line search

* Corresponding author.
Email addresses: alopez-o@uwaterloo.ca (Alejandro López-Ortiz), schuiere@informatik.uni-freiburg.de (Sven Schuierer).

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

problem consists of an agent searching for a target in an unknown terrain. In the worst case a search by a robot in a general domain can be arbitrarily inefficient as compared to the shortest path from the initial position to the target. However, as it is to be expected, strategies can be improved depending on the type of terrain and the searching capabilities of the robot.

In this work we assume that the robot is equipped with an on-board vision system that allows it to see its local environment. 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. As such, the performance of an on-line search strategy can be measured by comparing the distance traveled by the robot with the length of the shortest path from the starting point s to the target location t. The worst case 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.

There are several classes of polygons that admit constant competitive ratios, most notably *streets* [10,13,23], *$\mathcal{G}$-streets* [6,18] and *HV-streets* [5]. However, the existence of a searching strategy with a constant competitive ratio for these classes of polygons depends on the position of the target.

A natural question is to find non-trivial classes of polygons that allows searches with a constant competitive ratio independently of the starting position of the robot or the position of the target. Since the target may hide anywhere inside the polygon, a natural choice is to explore the class of polygons which can be seen in its entirety from a single point. Such a polygon is called *star-shaped*. The set of points that see the whole polygon is called the *kernel* of the polygon.

In a preliminary version of this paper [19] star-shaped polygons were the first class shown to admit position-independent target searches. Using techniques derived from this work, street polygons were later also shown to be searchable without restriction on the initial positions of the target or the searcher [4].

Icking and Klein studied the problem of on-line searching for the *kernel* of a star-shaped polygon. In this case, the competitive ratio is given by the ratio of the length of the path traversed by the robot from the starting point s to the closest kernel point p as compared to the distance from s to p. They present a strategy with a competitive ratio of ~ 5.331 [8,9], which was later shown to be exactly $\pi + 1$ competitive [16]. A strategy with a competitive ratio bounded by $1 + 2\sqrt{2} \sim 3.829$ was given by J-H. Lee et al. [17] and recently improved to ~ 3.1226 by L. Palios [21].

In this work we present a position-independent target search strategy with constant competitive ratio for star-shaped polygons. In section 2 we introduce some concepts and definitions for on-line searching in simple polygons. In section 3 we present a 11.52-competitive algorithm for target searching in star-

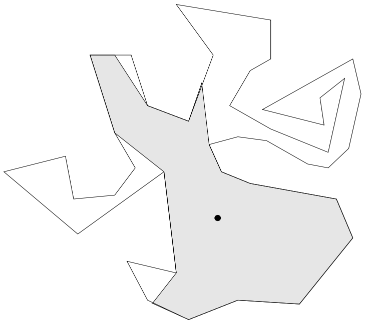

Fig. 1. Visibility polygon.

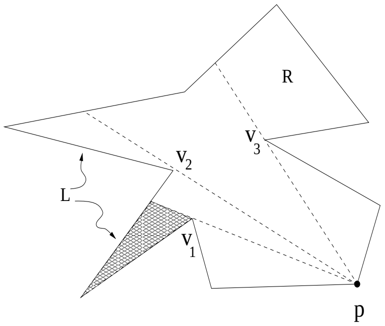

Fig. 2. Left and right pockets.

shaped polygons and prove a lower bound of 9 for the competitive ratio of all search strategies in star-shaped polygons. In section 4 we use a variation of the strategy in section 3 to construct the first constant competitive algorithm for recognition of star-shaped polygons. That is, given a polygon, the robot follows a path that proves or disproves that the polygon is star-shaped that is no more than 28.85 times longer than a shortest path with the same property. We present a lower bound of $\sqrt{82}$ for this problem.

Finally, we improve the lower bound of $\sqrt{2} \sim 1.41$ for searching for a kernel of a star-shaped polygon [8] first to ~ 1.49 and then to ~ 1.5.

2 Definitions

We say two points p and p' in a polygon P are visible to each other if the line segment $\overline{pp'}$ is contained in P. If A and B are two sets, then A is weakly visible from B if every point in A is visible from some point in B. The visibility polygon of p is the subset of points in P that are visible to p; it is denoted by $V_P(p)$. We assume that the robot has access to its local visiblity polygon by a range sensing device, e.g. a laser radar (also known as ladar). Now we can define a star-shaped polygon.

Definition 1 ([22]) *A simple polygon P is a* star-shaped polygon *if there exists a point p in P such that $V_P(p) = P$. The set of all points p inside P with $V_P(p) = P$ is the* kernel *of P.*

If the robot does not start in the kernel of P, then there are regions in P that cannot be seen by it. The connected components of $P \setminus V_P(p)$ are called pockets. The boundary of a pocket is made of some polygon edges and one line segment that does not belong to the boundary of P—which is called a window of $V_P(p)$. Note that a window intersects the boundary of P only in its end points, one of the endpoints is a vertex of P. This point is called the entrance to the pocket. More generally, a line segment that intersects the boundary of P only in its end points is called a chord.

Let p and q be two points in P. We denote the shortest path from p to q by $shp(p, q)$. The union of all shortest paths from p to the vertices of P forms a tree-like structure called the shortest path tree of p. We denote it by $shp\text{-}tree(p)$. Such a structure can be enlarged by prolonging each line segment of the shortest path tree away from p until it intersects the boundary of P. We term this new structure the extended shortest path tree of p which we denote by $shp\text{-}tree^*(p)$. The end points of the prolonged line segments are called the extension points of p.

The extended shortest path tree of p partitions P into triangles. The point of a triangle that is the closest to p is called the anchor of the triangle.

A pocket edge of p is a line segment (in P) that starts at p and contains at least one window. Each pocket edge is part of the extended shortest path tree $shp\text{-}tree^*(p)$ of p. More generally, an extended pocket path of p is a shortest path from p to an extension point of p. Obviously, an extended pocket path is also part of $shp\text{-}tree^*(p)$.

A pocket is said to be a left pocket if it lies locally to the left of the pocket ray that contains its window. A pocket edge is said to be a left pocket edge if it defines a left pocket. An extended pocket path is a left extended pocket path if its first line segment is collinear with a left pocket edge. Right pocket, right pocket edge, and right extended pocket path are defined analogously.

Since a point in the kernel of P sees all the points in P, in particular p, a pocket of $V_P(p)$ does not intersect the kernel of P which implies the following observation.

Observation 1 *The kernel lies locally to the right of all left pocket edges and locally to the left of all right pocket edges.*

For example, in the polygon of Figure 2, the kernel, if it exists, lies to the right of the pocket edges $\overrightarrow{pv_1}$ and $\overrightarrow{pv_2}$ and to the left of $\overrightarrow{pv_3}$.

Observation 1 implies that, in a star-shaped polygon, all left pocket edges appear consecutively in a clockwise scan of the boundary; similarly, all right pocket edges appear consecutively in a counterclockwise scan of the boundary. In particular, we can order the left pocket edges clockwise and the right pocket edges counterclockwise before we switch from left to right pocket edges or vice versa. Hence, there is a clockwisemost (or rightmost) left pocket edge E_l and a counterclockwisemost (or leftmost) right pocket edge E_r. The kernel is between E_l and E_r. We will make use of this ordering in our algorithm to search in a star-shaped polygon. Note that we can extend this ordering to extended pocket paths as well.

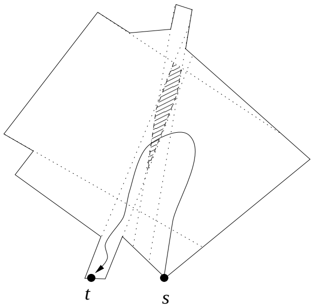

Fig. 3. Searching for a target via the kernel.

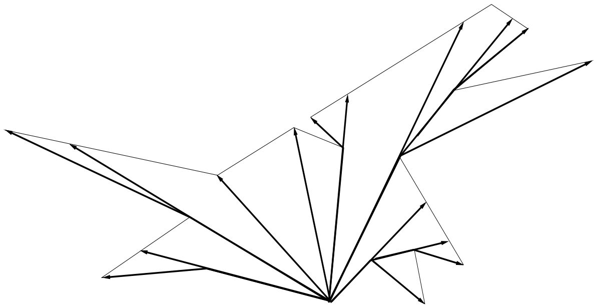

Fig. 4. Extended pocket path search.

3 Target Searching in Star Polygons

In this section we present a strategy to search for a target in a star-shaped polygon. At first sight it may seem that searching for a target is a variation of searching for the kernel of P: A natural strategy is to first search for the kernel and once it is reached to go directly to the target—since the target is seen from any point in the kernel. However, as illustrated in Figure 3 this approach does not lead to a constant competitive ratio.

In fact, searching for a target of unknown location inside a star-shaped polygon provably requires a larger detour in the worst case than walking into the kernel. We show this in the second part of this section. First we present a strategy to search for a target in a star polygon. We start with the following observation.

Lemma 1 *If c is a chord in star polygon P that splits P into two parts P_1 and P_2, then one of P_1 and P_2 is weakly visible from c and the other contains at least one point of the kernel of P.*

Proof: Let q be a point in the kernel of P. The point q is contained in one of the two parts, say in P_1. As q is in the kernel, all of P_2 can be seen from it. But any line contained in the polygon and joining a point in P_1 with a point in P_2 intersects the chord c. This implies that the chord weakly sees all points on the opposite side as well. □

Notice that the above lemma also holds for a simple path joining two points on the boundary of the polygon.

Theorem 1 *There exists a strategy for searching for a target inside a star-shaped polygon with a competitive ratio of at most 14.5.*

Proof: In the following we describe a strategy to search in a star-shaped polygon. The main idea is to search alternatingly on a (suitably chosen) left and right extended pocket path from s increasing the search depth each time by a constant factor c. For the algorithm we need the definition of a "rightmost" left extended pocket path, which we denote by E^d_{left}, for a given distance d. The idea is that we want to seal off as much as possible to the left of E^d_{left} if we go a distance of at most d. To this end let $\mathcal{V}$ be the set of all reflex vertices of P and extension points of the start point s of the search; let $v \in \mathcal{V}$ be a point belonging to a left extended pocket path E_v and similarly, let $v' \in E_{v'}$ where $E_v \neq E_{v'}$. We say that $shp(s, v)$ is to the right of $shp(s, v')$ if either $shp(s, v)$ contains $shp(s, v')$ or E_v is to the right of $E_{v'}$ (see Figure 4).

Let v be a point in $\mathcal{V}$ such that $shp(s, v)$ is the rightmost shortest path with length at most d. If such a path exists, then the extended pocket path E^d_{left} is defined as $shp(s, v)$ together with the extension of the last line segment of $shp(s, v)$. If such a path does not exist, then we arbitrarily define E^d_{left} to be the rightmost left extended pocket path. It is easy to see that E^d_{left} can be computed on-line by repeatedly going to the rightmost visible point $v \in \mathcal{V}$ on a left extended pocket path such that $d(s, v) \leq d$. The definition of a "leftmost" right extended pocket path E^d_{right}, for a given distance d, is analogous. This is illustrated in Figure 4, the value of d in E^d_{right} in this example is such that $d(s, u) \leq d < d(s, u')$.

Finally, we define $\neg left = right$ and $\neg right = left$. The algorithm can now be described as follows.

```
Algorithm Star Search
Input:  A star polygon $P$ and a starting point $s$
Output The location of the target point $t$
    let $p_0$ be the closest entrance point to $s$ and $E_0$ the pocket edge
    corresponding to $p_0$
    let $d_0 \leftarrow d(p_0, s)$; let $i \leftarrow 0$
    if $E_0$ is a left pocket edge
      then let $side \leftarrow left$
      else  let $side \leftarrow right$
    loop
      traverse $d_i$ units on $E_i$ starting from $s$
      if $t$ is seen then exit loop
      let $d_{i+1} \leftarrow c \cdot d_i$
      move back to $s$
      let $side \leftarrow \neg side$
      let $E_{i+1} \leftarrow E^{d_{i+1}}_{side}$
      $i \leftarrow i + 1$
    end loop
    if $t$ is seen then move to $t$
```

In the following we show that when the algorithm terminates, it has seen the target, and it travelled no more than 14.5 times the distance from s to t.

We first show that after the first two iterations the loop has the following invariant on line 11:

> Invariant: *All triangles of the partition induced by shp-tree*$^*(s)$ *whose anchor has a distance of at most* d_{i-2} *to* s *have been explored.*

Proof: (Invariant) Assume that *side* = *left* and consider a triangle T of the partition induced by *shp-tree*$^*(s)$ whose anchor v has a distance of at most d_{i-2} to s. We assume that the vertex v belongs to a left pocket. The argument is analogous with E_{i-1} instead of E_{i-2} if v belongs to a right pocket.

If v belongs to E_{i-2}, then, clearly, T has been explored. If v does not belong to E_{i-2}, then E_{i-2} is to the right of *shp*(s, v). In particular, there is a point v' on E_{i-2} with $v' \in \mathcal{V}$ and $d(s, v') \leq d_{i-2}$ such that v is to the left of *shp*(s, v'). Hence, *shp*(s, v') contains a chord c such that T belongs to the subpolygon to the left of c. Since the kernel of P is the right of c by Observation 1, T is weakly visible from c by Lemma 1.

The correctness of the algorithm now follows from the fact that d_i increases exponentially and all the triangles that belong neither to a left nor a right pocket are visible to s.

We claim that Algorithm *Star Search* has a competitive ratio of 14.5. The worst case to discover the target occurs when the robot sees the target at a distance of $d_i/c^2 + \epsilon$, at the very end of Step i when it has traveled a distance of d_i (see Figure 5; notice that q_i defines the entrance to a left pocket). The distance traveled by the robot to go to t is now $2d_0 \sum_{j=0}^{i-1} c^j$ for the previous steps, $d_0 c^i$ to discover t, $d_0 c^i$ to return to s, and $d(s, t)$ to go to t. Hence, the competitive ratio is bounded by

$$\frac{d(s,t) + 2\sum_{j=0}^{i} c^j d_0}{d(s,t)} \leq 1 + 2\frac{\sum_{j=0}^{i} c^j}{c^{i-2}} \leq 1 + \frac{2c^3}{c-1}.$$

Substituting the value 3/2 which minimizes $2c^3/(c-1)$ gives a competitive ratio of 14.5. In fact, it can be shown that there is no choice of the step lengths d_i that yields a better competitive ratio for the above algorithm [1,7]. □

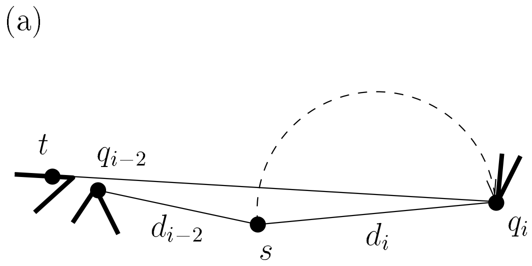

Fig. 5. The worst case to discover the target. If the robot follows the dashed path, then t is detected at w instead of q_i.

3.1 Improving the Strategy

Let q_i be the end point of the exploration of E_i visited in Step i. The worst case configuration occurs when the angle $\angle q_{i-2}sq_i$ is relatively flat. In this case the competitive ratio can be improved if the robot does not follow the straight line segment $\overline{sq_i}$ but follows a curve that allows it to detect the target earlier (see Figure 5). So instead of traveling along one chord $f_\ell = \overline{v_\ell v_{\ell+1}}$ that belongs to E_i the robot now travels along the semi-circle C_ℓ that is spanned by f_ℓ.

More precisely, the robot computes the curve $\mathcal{C}_\ell$ that is the upper envelope of all circles that are contained in the polygon and whose diameter is contained in f_ℓ. The curve $\mathcal{C}_\ell$ consists of parts of circles $C^{(1)}, \ldots, C^{(k_\ell)}$. The center $c^{(j)}$ of each circle $C^{(j)}$ is contained in f_ℓ with $c^{(j+1)}$ to the right of $c^{(j)}$, for $1 \leq j \leq k_\ell - 1$. An equivalent, constructive, definition is as follows. $C^{(1)}$ is the maximal inscribed circle passing through v_ℓ with center on f_ℓ and inscribed by $P \cup \{v_{\ell+1}\}$. If this half circle reaches $v_{\ell+1}$ this completes the construction of $\mathcal{C}_\ell$. Otherwise the half circle contains a point $q^{(1)}$ on the boundary of P. By construction, the entire half circle is contained in P. Hence $q^{(1)}$ is visible from v_ℓ. To construct $C^{(1)}$ in this case, the robot examines all visible points (from v_ℓ) on the boundary of P and identifies the point of inscription $q^{(1)}$ which defines the center and radius of $C^{(1)}$. The part of $C^{(1)}$ between v_ℓ and $q^{(1)}$ is the first part of $\mathcal{C}_i$. Now assume that $\mathcal{C}_i$ is already constructed up to circle $C^{(j)}$ with $1 \leq j \leq k_i - 1$. There is a point $q^{(j)}$ such that $C^{(j)}$ intersects the boundary of P in $q^{(j)}$. The circle $C^{(j+1)}$ is defined as a maximal-radius half circle with center and diameter on f_ℓ inscribed inside P passing through $q^{(j)}$ (see Figure 6). An edge e incident to $q^{(j)}$ may intersect all circles with center to the right of $c^{(j)}$ that contain $q^{(j)}$. In this case a circle C with center to the right of $c^{(j)}$ is computed that intersects one end point v of e and a different edge e' of the boundary of P. We set $q^{(j+1)} = v$. The part of e from $q^{(j)}$ to $q^{(j+1)}$ is included in $\mathcal{C}_i$ and, for simplicity, denoted as $C^{(j+1)}$. The circle $C^{(j+2)}$ is now defined as C and the intersection point of C and e' is the point $q^{(j+2)}$.

Note that we can use other curves in the above construction as well. A semi-

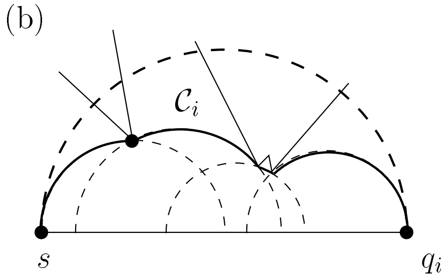

Fig. 6. The new strategy of the robot.

circle is the set of points p such that the angle θ at p between the line segments connecting p to the left and right end point of the semi-circle is $\pi/2$. Instead of choosing $\pi/2$ we may also choose other values for θ. We again obtain circle segments in this way. The construction of $\mathcal{C}_\ell$ is also well defined using the new type of curves. In the appendix it is shown that the length of $\mathcal{C}_\ell$ is at most $(\pi - \theta)/\sin\theta\ d(v_\ell, v_{l+1})$.

3.1.1 Analysing the Competitive Ratio

Now assume that n is the last step of Algorithm *Star Search* and that *side* = *left*. We analyse the distance that the robot travels in this step. First assume that E_n is a simple pocket edge. Let $C^{(j)}$ be the circle of $\mathcal{C}_n$ which separates t from s. Let q_ℓ be the left point and q_r the right point at which $C^{(j)}$ intersects E_n. Furthermore, let p be the point at which the robot detects t, B_n be the arc of $C^{(j)}$ from p to q_r, and $\mathcal{C}_p$ the part of $\mathcal{C}_n$ from s to p. By our above considerations the length of the concatenation of $\mathcal{C}_p$ with B_n is bounded by $(\pi - \theta)/\sin\theta\ d(s, q_r)$. Hence, if we consider the path $\widetilde{\mathcal{C}}$ that consists of the concatenation of the circle segment C over $\overline{sq_\ell}$ and $C^{(j)}$, then this path is at least as long as $\mathcal{C}_p$ concatenated with B_n. Since B_n is also part of $\widetilde{\mathcal{C}}$, the part of $\widetilde{\mathcal{C}}$ from s to p is at least as long as $\mathcal{C}_p$. In the following we assume that the robot has travelled along $\widetilde{\mathcal{C}}$ instead of $\mathcal{C}_p$ (see Figure 7). We choose a coordinate system with origin at q_ℓ and $\overline{q_\ell q_r}$ as the unit vector in x-direction. We first show how to reduce the case that t is to the left of y-axis to the case that t is on or to right of the y-axis. If t is to the left of the y-axis, then let q be the intersection point of the path traveled by the robot with the y-axis. There is a vertex v on $C^{(j)}$ that belongs to $shp(s,t)$. Let $\mathcal{C}[q]$ be the part of the path of the robot from s to q.

Assume that we can show that $|\mathcal{C}[q]| \leq c\, d(s,q) = c\, d(s,v) + c\, d(v,q)$, for some $c > 1$. If we observe that the angle $\angle tqv$ is at least $\pi/2$ since the line through $\overline{qt}$ is collinear with q_r, then simple trigonometric calculations[2] show that the length of the path of the robot is bounded by

[2] If we choose a coordinate system with origin at v and $\overline{vq}$ as the unit vector in

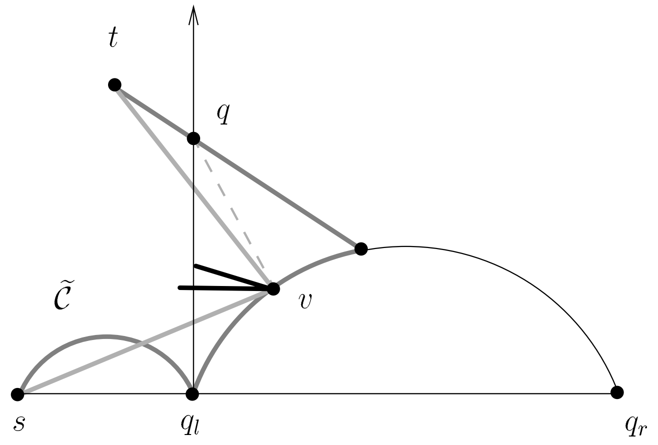

Fig. 7. If $\mathcal{C}_n$ consists of more than one circle, then it can be assumed that the robot first travels from s to q_l.

$$
\begin{aligned}
|\mathcal{C}[q]| + d(q,t) \leq cd(s,v) + cd(v,q) + d(q,t) \quad &\leq \; cd(s,v) + \sqrt{c^2+1}d(v,t) \\
\leq \sqrt{c^2+1}d(s,t) &
\end{aligned}
$$

Hence, we only need to analyse the case that t is on or to the right of the y-axis.

We assume that t has coordinates $(x,1)$, $x \geq 0$, and s has coordinates $(-z,0)$. We now compute what distance the robot traverses in the last step. It first travels a distance of at most $(\pi-\theta)/\sin\theta\; z$ to reach q_ℓ. Now consider how much the robot travels from q_l to t. The situation is displayed in Figure 8.

Note that the robot detects t at p at the latest; otherwise, t is not seen by $\overline{q_\ell q_r}$ in contradiction to Lemma 1. We denote the angle $\angle pq_\ell t$ by β and the angle $\angle q_r pt$ by α. Some simple trigonometric calculations show that the ratio of the

x-direction and let $\alpha = \angle tqv$, then

$$d(v,t) \geq \sqrt{(1+x\cos(\pi-\alpha))^2 + x^2\sin^2(\pi-ga)}$$

where $x = d(q,t)$. We are interested in the smallest $d > 0$ such that

$$c + x \leq d\sqrt{(1+x\cos(\pi-\alpha))^2 + x^2\sin^2(\pi-ga)}$$

The right hand side is minimized for $\alpha = \pi/2$ and the inequality simplifies to

$$c + x \leq d\sqrt{1+x^2} \qquad \text{or} \qquad d^2(1+x^2) - c^2 - 2cx - x^2 \geq 0.$$

If we minimize w.r.t. x, we obtain $x = c/(d^2-1)$ and the above condition simplifies to $d^2 - c^2 \geq c^2/(d^2-1)$ which leads to $d \geq \sqrt{c^2+1}$.

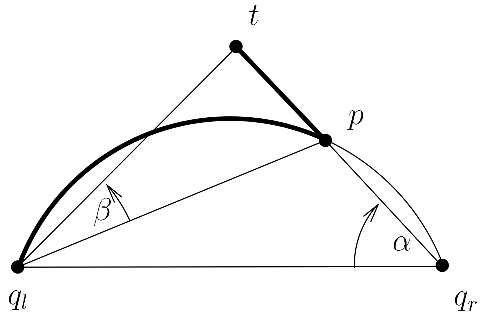

Fig. 8. Bounding the distance traveled by the robot if E_n is a simple pocket edge and $\mathcal{C}_n = C_n$.

distance traveled by the robot to $d(s,t)$ is bounded by

$$\frac{(\pi-\theta)\sin(\theta-\beta)}{\sin^2\theta} + \frac{\sin\beta}{\sin\theta}.$$

If we set $\theta = 2.152$, then the maximum for the above expression is assumed for $\beta \sim 1.03$ and is less than 2.302.

Since $d(q_r,t) \geq \sqrt{x^2+1}$ and $d(s,t) = \sqrt{(x+z)^2+1}$, the ratio of the distance traveled in the last step to $d(s,t)$ is given by

$$\frac{(\pi-\theta)/\sin\theta z + c\sqrt{x^2+1}}{\sqrt{(x+z)^2+1}}$$

where $c = 2.302$. If we maximize the above expression w.r.t. to z, then we obtain a function which is convex in x and a simple calculation shows that the maximum is achieved for $x = 0$. This leads to a value of ~ 2.588 and the competitive ratio is bounded by

$$\frac{\left(1+\frac{\pi-\theta}{\sin\theta}\right)\sum_{i=0}^{n-1} c^i d_0 + (\sqrt{2.59^2+1})d(s,t)}{d(s,t)} \leq 2.78 + \left(1+\frac{\pi-\theta}{\sin\theta}\right)\frac{c^2}{c-1}$$

since $d(s,t) \geq c^{n-2}d_0$ and the distance that the robot travels on its way to q_i and back is bounded by $1 + (\pi-\theta)/\sin\theta \ d_i$ with $\theta = 2.152$. The above expression is minimized for $c = 2$ and we obtain that the competitive ratio is bounded by $2.78 + (1 + 1.1841) \cdot 4 < 11.52$. Note that worst case still occurs when the target is found in step i at a distance d_{i-2}. Indeed, in this case the robot traverses the entire path of step $i-1$ and back to the starting point s. Any distance traversed in step i only worsens the competitive ratio.

In the above analysis we have assumed that E_n is a simple pocket edge. If E_n is not a simple pocket edge, then the robot constructs the curve $\mathcal{C}_i$ described above for each edge of E_n separately. Note that the circles constructed for a left pocket edge are to the left of E_n and, therefore, the circles over one edge of E_n are naturally separated from the circles over another edge by the vertices of the polygon that are intersected by E_n. Moreover, the situation if t is detected to the left of the edge e of E_n, then we can consider the closer end point p of e to be s and apply the above analysis. Since the shortest path from s to t also goes through p, the competitive ratio is only decreased by adding the path from s to p. We have shown the following result which improves Theorem 1.

Theorem 2 *There exists a strategy for searching for a target inside a star polygon with a competitive ratio of at most 11.52.*

It is interesting to note that the above algorithm can also be used as algorithm to "look around a corner" in a simple polygon [11].

3.2 A Lower Bound on the Competitive Ratio

In this section we show that any on-line strategy to search in a star-shaped polygon has a competitive ratio of at least 9. Our lower bound proof is based on a reduction to a variant of *searching on the real line*. In this setting the robot starts at the origin of the real line and has to find a target that is located either to its left or to its right. The distance to the target is at least one. The robot can only detect the target if it stands on top of it.

Before we present the reduction, we first need to argue about some properties of strategies to search on the real line. To start with we need the following theorem.

Theorem 3 ([20]) *Any on-line strategy to search on the real line for a target at a distance of at most D has a competitive ratio of at least $9 - f(D)$ where $f(D) \leq 162/\log^2(D/16)$, for sufficiently large D.*

Now that we have a precise bound on searching on a line segment, we can use this fact to obtain a lower bound for target searching in a star-shaped polygon.

Theorem 4 *There exists a family of star-shaped polygons such that the worst case competitive ratio for finding a target of unknown position has a competitive ratio of at least 9.*

Proof: Consider the polygon in Figure 9. The length of the base is $2n$ and the height is $n^4 + 1$. The base of each of the indentations is of width one-half and spaced one-half units apart. Let the center of the base be the origin. For each dent one of its lower corners lies exactly over a point of integer coordinates

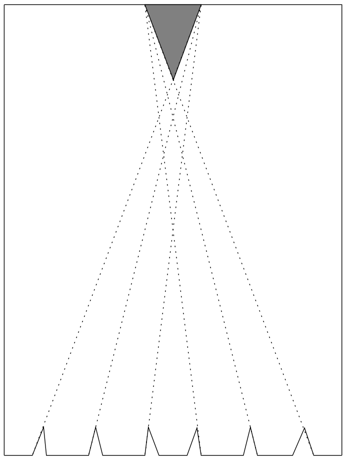

Fig. 9. Lower bound for searching for a target.

on the x axis. The slope of a wall at position i is $-n^4/i$. The intersection of all the half-planes in the polygon contain the point $(0, n^4)$, thus forming a start-shaped polygon.

Now, the robot follows a search path S in the plane. This path S must intersect the extension of the quasi-vertical walls of each dent to "see" into each dent until it finds the target and moves towards it. Without loss of generality, the strategy S visits some quasi-vertical extended lines on the positive side of the x-axis up to wall x_1, then turns around and eventually examines some extended lines on the negative side of the x-axis up to wall x_2, then on the positive side until x_3, and so on until all dents have been examined.

The sequence $X = (x_1, x_2, \ldots, x_m)$ also describes a strategy for searching on the real line, namely the strategy that moves from the origin to x_1, then back to the origin and past to x_2, then to x_3 and so on. As we know any such strategy is $9 - f(n)$ competitive.

Let x_i be a dent where X is $9-f(n)$ competitive. Let $p_i = (u_i, v_i)$ be the point where the robot path first intersects the extension of the quasi-vertical line over x_i. Note that $v_i = (x_i - u_i)\, m_i$ where $m_i = -n^4/i$. Therefore if $v_i > 9\,n$ the competitive ratio is at least $9\,n/x_i > 9$ and there is nothing to show.

Otherwise, the distance traversed by the robot from the quasi-vertical line above x_i to x_{i+1} is given by $d(p_{i+1}, p_i) \geq |u_{i+1} - u_i| = |u_{i+1}| + |u_i|$. If x_i is positive $u_i > x_i - 9\,n/m_i > x_i - 9/n^2$ and otherwise $u_i < x_i - 9\,n/m_i < x_i + 9/n^2$ (recall that $v_i \leq 9\,n$). Therefore $|u_{i+1} - u_i| \geq |x_i - 9/n^2| + |x_{i+1} + 9/n^2|$ for positive x_i, and similarly $|u_{i+1} - u_i| \geq |x_i + 9/n^2| + |x_{i+1} - 9/n^2|$ for negative x_i. In both cases we have $d(p_i, p_{i+1}) > |x_i| + |x_{i+1}| - 18/n^2$ which implies that the competitive ratio of S is

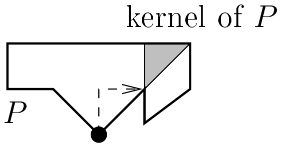

Fig. 10. Lower bound configuration for walking into the kernel.

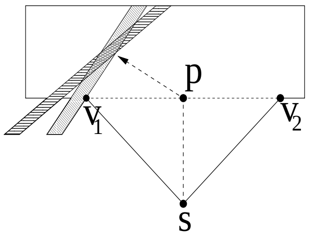

Fig. 11. Polygon with two beams.

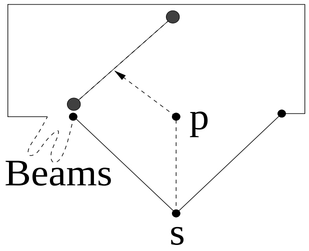

Fig. 12. Lower bound configuration.

$$
\begin{aligned}
C_S &\geq \frac{\sum_{j=1}^{i} d(p_j, p_{j+1}) + d(p_{i+1}, p_i) + d(p_i, x_i)}{|x_i|} \\
&\geq \frac{\sum_{j=1}^{i}(|x_j| + |x_{j+1}| - 18/n^2) + |x_i| + |x_{i+1}| - 18/n^2}{|x_i|} \\
&= 9 - f(n) - \frac{18(i+1)}{|x_i|\, n^2} \\
&\geq 9 - f(n) - \frac{18}{n^2}
\end{aligned}
$$

This last expression converges to 9 as $n \to \infty$. □

4 Walking into the Kernel—a Lower Bound

In this section we consider the problem of on-line searching and walking into the kernel of a star-shaped polygon [8,9,17]. We present a lower bound of ~ 1.492 on the competitive ratio of any strategy to walk into the kernel.

Figure 10 shows a lower bound of $\sqrt{2}$. Any on-line strategy with a competitive ratio of at most $\sqrt{2}$ has to follow the dashed path [8]. In the following we show that any strategy to search for the kernel of a star-shaped polygon has a competitive ratio that is significantly larger than $\sqrt{2}$.

Definition 2 *The visibility region of a subset B of a polygon is the set of all points in the polygon which see all points in B.*

Definition 3 *Given the current position of the robot p and a pocket B of $V_P(p)$, the* beam *of the pocket is the visibility region of B.*

Notice that if the pocket is a trapezoid, the visibility region resembles a search light beam (see Figure 11).

Observation 2 *The kernel lies in the intersection of all beams.*

Lemma 2 *Walking into the kernel of a polygon is at least* 1.492*-competitive.*

Proof: Consider the polygon of Figure 11. Notice that the robot reaches the line segment $\overline{v_1 v_2}$ before it reaches the kernel. In addition, it is favourable for the robot to reach $\overline{v_1 v_2}$ at its midpoint p, as otherwise the following construction can be made on the side opposite to the robot's preferred side which obviously only increases the competitive ratio of the strategy. From p it is not yet clear where the kernel is located. In fact, depending upon the specific angle and location of the pockets, the beams might specify a small kernel located anywhere in the visibility polygon region of s which is above $\overline{v_1 v_2}$.

We now use an adversary argument. After the robot reaches p the adversary closes one side, and selects two candidate kernels, illustrated by the large dots in Figure 12, such that one is next to v_1 and the other on the line through s and p. This can be achieved by locating one beam A along the line joining the two candidate regions, and a second beam, B, nearly parallel and to the right of A. The intersection of both beams defines the kernel of the polygon.

We assume that the robot learns of this decision and, thus, can restrict itself, to its benefit, to determining which of the two regions is the kernel.

Still, the robot cannot decide which of the two candidates is the kernel before it reaches at least one of the beams A or B. If we choose the beams to have width $\varepsilon > 0$ and to be ε apart, then the robot has to come within a distance of $5/2\varepsilon$ or less of the line ℓ joining the midpoints of the two candidate regions.

By some simple trigonometric calculations (see Appendix B) one can show that if we choose the angle α of the line ℓ with the horizontal axis to be 0.655, then any strategy generates a path with a competitive ratio of at least 1.492 (if we choose ε sufficiently small). □

Remark 1 *Notice that in fact the adversary can pre-select n candidate slopes instead of just one. Once again, we assume that the robot learns of this decision and chooses a strategy that optimizes the competitive ratio for the candidate slope set. In particular, if the adversary chooses three slopes with angles* $0.51, 0.65$ *and* 0.82 *radians, then numerical calculations show a lower bound of at least 1.5 on the competitive ratio of the robot.*

5 Recognition of Star-Shaped Polygons

For the on-line star-shaped recognition problem, we assume that given a polygon P, it is the robot's task to determine if P is star-shaped. The competitive ratio is now given by the ratio of the distance traversed by the robot to the

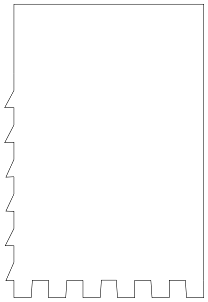

Fig. 13. Lower bound for recognition

length of the shortest path that proves or disproves that a given polygon is star-shaped. We present a strategy that recognizes a polygon at a constant competitive ratio, both in positive and negative instances, provided that the shortest path that recognizes the polygon is of length greater than or equal to a fixed $\epsilon \geq 0$. Furthermore, if the polygon is star-shaped the proposed strategy reaches the kernel, if it exists, at a constant competitive ratio as well.

Theorem 5 *Recognition of a star polygon is at least $\sqrt{82}$ competitive.*

Proof: Consider the polygon in Figure 13. This polygon is of similar dimensions and configuration to the polygon of Figure 9, with the addition of dents on the left vertical edge of the polygon, which are also half-unit sized and half-unit spaced. The dents at the base are called *base dents* and the dents on the left vertical wall are called *side dents.* The side dents are formed of two edges, one horizontal, the other of slope $(n^4 + 1 - i)/n$ for a dent of height i. This ensures that the intersection of all the half-planes in the polygon still contain the point $(0, n^4)$, thus forming a star-shaped polygon. Alternatively, the horizontal edge of any given side or base dent might contain a small spiral making the polygon not star-shaped.

The robot search path must intersect the extension of the quasi-vertical walls of each base dent and the extension of the horizontal edge of side dents to "see" into each dent until either finds a spiral thus proving that the polygon is not star-shaped, or it has examined all the dents proving that the polygon is indeed star-shaped.

As in Section 3.2 we consider the sequence $X = (u_1, u_2, \ldots, u_m)$ of base dents examined by the robot. This sequence describes a strategy for searching on the real line which is at least $9-f(n)$ competitive. Let $p_j = (u_j, v_j)$ be the point of first intersection of the robot path with the extension of the quasi-vertical line over x_j. Let w_i be a dent where the strategy X reaches the $9-f(n)$ competitive ratio. Let x_i be the base dent corresponding to w_i and let $d = |x_i|$. Note that $v_i = (x_i - u_i)\, m_i$ where $m_i = -n^4/i$. If $v_i > \sqrt{82}\, n$ the competitive ratio is at least $\sqrt{82}\, n/x_i \geq \sqrt{82}$ and there is nothing to show.

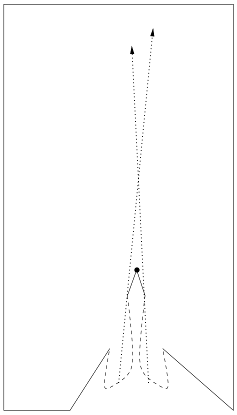

Fig. 14. Unboundedness of recognition

Now if the dent located at $-x_i$ has not yet been examined, then the competitive ratio is at least $((9 - f(n))|u_i| + d(p_i, p_{-i}))/d \geq ((9 - f(n))\, d + d)/d \geq 10 - f(n)$. Hence we assume that both base dents at distance d have been examined.

We now turn our attention to the side dent at height d. If it is unexplored, we let the strategy proceed until it has and call this path Γ_d, otherwise, the path up to p_i is Γ_d. The path Γ_d can be split into monotonous paths along the x and y axis. The sum of the length of the projection of the monotonous paths along the y axis is at least d and along the x axis is at least $(9 - f(n))\, u_i \geq (9 - f(n))(d - \sqrt{82}/m_i) \geq (9 - f(n))(d - \sqrt{82}/(n^4/d))$. Therefore the total distance traversed by Γ_d is at least $|\Gamma_d| \geq \sqrt{d^2 + (9 - f(n))^2(d - d\sqrt{82}/n^4)^2} \geq d\sqrt{82 - O(f(n))}$ which implies $C_S \geq |\Gamma_d|/d \geq \sqrt{82 - O(f(n))} \to \sqrt{82}$ as $n \to \infty$. □

We now give a strategy for recognizing star-shaped polygons. In general this problem is of unbounded competitive ratio. Indeed, let P be a simple polygon. We denote by Γ_{opt} the shortest path that decides whether P is star-shaped or not. In the case of the polygon of Figure 14, reaching either of the two dotted lines might result in a spiral being found and thus the polygon is rejected. However the robot cannot determine from the available information on which side is the closest dotted line and at what distance. Therefore if the robot moves, say, to the right for a distance ϵ, we place the left dashed line at a distance ϵ^2 resulting in a competitive ratio of $1/\epsilon$. Thus we require that $|\Gamma_{opt}| \geq \epsilon$ for some fixed, known ϵ. The path Γ_{opt} also has the following properties.

Definition 4 *The visibility polygon of a set A inside a polygon P is defined as $V_P(A) = \cup_{x \in A} V_P(x)$.*

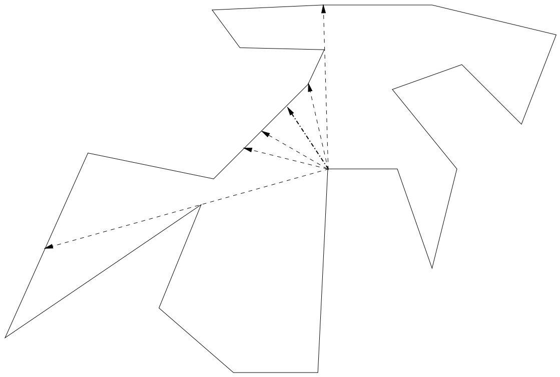

Fig. 15. Local left pocket edges.

The following lemma states that the robot cannot infer the star-shapedness of the polygon from information beyond that obtained from the visibility polygon. This is not immediately obvious, as in principle, the robot could, for example, attempt to use the location of the pockets to infer the nature and potential location of the edges inside them and deduce that the polygon could never be completed into a star-shaped one.

Lemma 3 *If P is star-shaped, then Γ_{opt} is the shortest path that sees all the points in P and if P is not star-shaped, then Γ_{opt} is the shortest path in P such that the visibility polygon of Γ_{opt} is not star-shaped.*

Proof: The proof is by contradiction. So first assume that the robot accepts a star-shaped polygon without having looked into all of its pockets; then, an adversary creates a spiral in that pocket, and the polygon is not star-shaped—a contradiction.

On the other hand, assume that the robot rejects the polygon. Consider first the intersection A of the half planes defined by already seen edges of the polygon. These half planes are closed sets and their intersection forms a closed convex set in the plane. Secondly, consider the intersection B of the half planes defined by pocket edges into pockets that remain unexplored. These half planes are open, and thus their intersection is either the interior of a polygon or the empty set.

The kernel, if it exists, lies in the intersection of A and B. If the robot rejects the polygon P while $A \cap B \neq \emptyset$ then the adversary selects a point p in $A \cap B$, and "empties" all the pockets by means of inserting and almost flat two edge chain closing the pocket (say the chain is ϵ dented by a vertex on its midpoint). Because B is an open set, it follows that there exists small enough ϵ such that the intersection of all the open (and therefore closed) half-planes of this modified polygon contains p and thus the polygon is star-shaped—again a contradiction. □

We also need the following generalization of extended pocket paths.

Definition 5 *We say that a straight chord is a* local left pocket edge *if it is contained between two consecutive left extended pocket paths and one of its end points lies on the leftmost of the two pocket paths. The definition of a* local right pocket edge *is analogous.*

For convenience we consider in the following a local pocket edge together with the extended pocket path to which one of its end points belongs also as an extended pocket path.

In effect, this definition enlarges the set of extended pocket paths to include the shortest path tree to *edges* as well as vertices. The robot then explores this tree. Figure 15 illustrates a polygon with local left pocket edges. Notice that in this case some of these pockets are nearer to the starting point s than the entrance point of the preceeding left pocket edge. In this scenario, it is to the robot's advantage to explore along the local left pocket edge rather than on the preceeding left pocket edge.

Theorem 6 *There exists a* 28.85*-competitive strategy that identifies if a polygon is star-shaped or not.*

Proof: The algorithm is almost the same as the one proposed for target searching in Theorem 1 except for Steps 8 and 10–11 which changed as follows. Let $side \in \{left,\ right\}$ as before.

Step 8 If the intersection of the half planes induced by the edges of the visibility polygon is empty, then the algorithm rejects P.

If all pockets have been explored, that is, P is completely visible, then the algorithm accepts.

Otherwise the robot continues its exploration.

Steps 10–11 If the pocket edges on the opposite side are not exhausted, the robot changes side $side \leftarrow \neg side$. It follows the extended pocket path E_i for a distance of d_i to a point p; starting at p it sweeps an arc G_i of the geodesic circle of radius d_i with center s in the Euclidean sub-space defined by the interior of P towards the extended pocket paths on the opposite side. It follows G_i until it reaches the boundary of P.

If the robot reaches an extended pocket path on the opposite side, then it identifies the next extended pocket path E_{i+1} to be explored (as in Algorithm *Star Search*), backtracks along G_i to the point at distance d_i on E_{i+1}, sets d_{i+1} to $c\,d_i$, and proceeds to the next exploration step.

Otherwise, if it reaches the boundary of P without crossing an extended pocket path of the opposite side, then the robot follows the ray towards s until it intersects the previous geodesic arc G_{i-1} and then moves along G_{i-1} to the next extended pocket path E_{i+1}. In this case we set $d_{i+1} = d_i$. The extended pocket paths on the side where the robot started are now considered as explored.

If the pocket edges on the opposite side are exhausted, the robot follows the extended pocket path E_i for a distance of d_i to a point p; backtracks along G_i to the point at distance d_i on E_{i+1}, sets d_{i+1} to cd_i, and proceeds to the next exploration step.

Invariant: *The visibility region of the path explored thus far by the robot contains the visibility region of any path of length d_i/c or less.*

As an example consider Figure 16. The robot first explores E_1, then swipes in a clockwise direction until it hits the boundary. At this point it sets d_2 to $c\,d_1$ and backtracks on the same circular arc until it crosses E_2 (selected as in the algorithm of section 3). From there it explores E_2 and swipes again, this time counterclockwise.

In the following we show the correctness and analyze the competitive ratio of the strategy. We first argue that the strategy is correct.

Consider Steps 10–11 and assume that the robot walks clockwise on a geodesic arc from left to right. If the robot cannot reach a right extended pocket path, then this implies that the robot is blocked by a boundary point p of P. We claim that p lies between the rightmost left and the leftmost right extended pocket paths. To see this we first note that it cannot be between two right extended pocket paths as in this case the robot would have crossed at least one right extended pocket path. It also cannot be between two left extended pocket paths since in this case p is the end point of a chord that starts on a left extended pocket path which implies that p belongs to a local extended pocket path. Since local extended pocket paths are also considered to be extended pocket paths, the current geodesic arc would start at or to the right of p (since p is at a distance of at most d_i from s)—a contradiction.

As there are no pockets between the rightmost left and the leftmost right extended pocket path, p is visible from s. The robot moves towards s until the previous geodesic arc is reached and continues exploring the right side. In this case the robot has completed searching the left side. The situation on the right side is analogous.

After the ith iteration of Steps 10–11 the path of the robot divides the polygon in two parts one of which completely contains all points at distance at most $d_0 c^{i-1}$. Observation 1 now implies that the invariant is correct. Step 10 accepts or rejects when either a contradiction to the fact that P is star-shaped is found or the whole polygon has been explored. Since eventually $d_0 c^i$ is larger than the diameter of P, P either becomes completely visible or is rejected before. This concludes the proof of correctness.

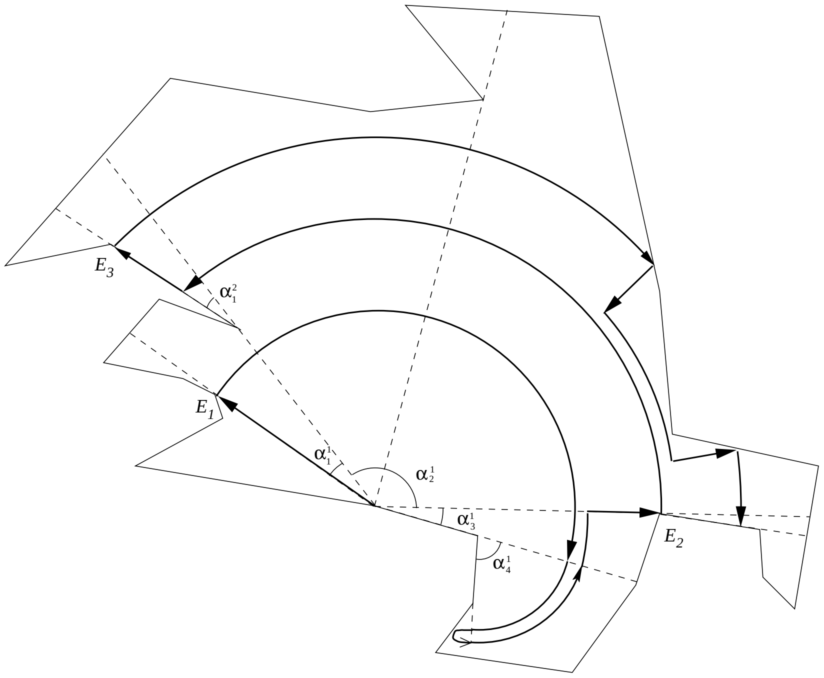

Fig. 16. Recognizing a polygon.

For the analysis we first observe that by the invariant and Lemma 3 the length of a shortest recognition path is at least $d_0 c^{k-1}$ if the algorithm stops after k iterations. In the following we determine an upper bound on the length of the path that is traversed up to and including the kth iteration of Steps 10–11 which yields immediately an upper bound on the competitive ratio.

We partition each geodesic arc G_i into two sets A_i and $\overline{A}_i$ as follows. The set A_i is defined as the set of the points p on G_i such that p is visible from s and the extension of the radius through p intersects G_{i+1} before it intersects the polygon border. The set $\overline{A}_i$ is defined as $G_i \setminus A_i$.

Similarly, we partition G_{i+1} in two sets B_{i+1} and $\overline{B}_{i+1}$. A point p belongs to the set B_{i+1} if there is a point q in A_i such that radius extension through q contains p. The set $\overline{B}_{i+1}$ is defined as $G_{i+1} \setminus B_{i+1}$. For example, in Figure 16, on the geodesic corresponding to E_1, the points contained in the arc corresponding to the angles α_1^1, α_3^1, and α_4^1 are in $\overline{A}_i$ while the points corresponding to angles α_2^1 are in A_i. For E_2 we have that the points in the arc over the angle α_1^2 are in $\overline{B}_2$ while the points corresponding to the angle $\alpha_1^2 = \alpha_2^2$ are in B_2.

Clearly, $\overline{A}_i$ consists of at most two (possibly empty) connected components. One of these components is to the left of A_i and the other to the right of A_i. Notice that one of these two subsets corresponds to the arcs that are backtracked when the robot moves on G_i to the $(i+1)$st extended pocket path E_{i+1}. Let α_i be the angle corresponding to A_i (and B_{i+1}), $\overline{\alpha}_i$ be the angle corresponding to $\overline{A}_i$, and $\overline{\beta}_i$ be the angle corresponding to $\overline{B}_i$. Note that the total angle γ_i of the arcs of G_i equals $\alpha_i + \overline{\alpha}_i$ and $\alpha_{i-1} + \overline{\beta}_i$. Clearly, since P is star-shaped, $\gamma_i \leq 2\pi$ as otherwise the intersection of the half planes determined by the edges which are enclosed by an extended pocket path is empty (in fact this holds for an arbitrary point s in the interior of the polygon).

The geodesic arc G_i consists of circular arcs which are centered at s or some vertex of P. We observe that the radius of a circular arc that belongs to $\overline{A}_i$ is at most d_0c^i and the radius of an arc that belongs to $\overline{B}_i$ is at most $d_0c^i - d_0c^{i-1}$. Hence, the distance traversed in step i, $i \geq 1$ at most

$$\overline{\alpha}_{i-1}\, d_0c^i + d_0c^i - d_0c^{i-1} + \overline{\beta}_i\,(d_0c^i - d_0c^{i-1}) + \alpha_{i-1}\, d_0c^i$$

where $\overline{\alpha}_{i-1}\, d_0c^{i-1}$ bounds the distance to backtrack on the $(i-1)$st geodesic to reach the ith extended pocket path E_i, $d_0c^i - d_0c^{i-1}$ is the distance to go from G_{i-1} to G_i and $\overline{\beta}_i\,(d_0c^i - d_0c^{i-1}) + \alpha_{i-1}\, d_0c^i$ bounds the length of the sweep on G_i.

Since $\gamma_i \leq 2\pi$, we have $\overline{\alpha}_i \leq 2\pi - \alpha_i$ and $\overline{\beta}_i \leq 2\pi - \alpha_{i-1}$. Hence, the distance traversed in step i, $i \geq 1$, can be bounded by

$$\begin{aligned}&(2\pi - \alpha_{i-1})d_0c^{i-1} + d_0c^{i-1} - d_0c^{i-1} + (2\pi - \alpha_{i-1})(d_0c^i - d_0c^{i-1}) + \alpha_{i-1}\, d_0c^i \\ &= d_0c^i - d_0c^{i-1} + 2\pi\, d_0c^i.\end{aligned}$$

Let k be the number of geodesics that are (partially) explored before the robot completes the recognition process. If the pockets on neither side are exhausted before the algorithm stops, then the total distance traversed is now bounded by the telescopic sum

$$\begin{aligned}d_0 + 2\pi\, d_0 + \sum_{i=1}^{k}\left[d_0c^i - d_0c^{i-1} + 2\pi\, d_0c^i\right] &= d_0c^k + 2\pi\, d_0 \sum_{i=0}^{k} c^i \\ &= d_0c^k + 2\pi\, d_0 \frac{c^{k+1}-1}{c-1}.\end{aligned}$$

Now consider the case when the pockets on one side are exhausted as illustrated in Figure 16. In this case, after the robot hits the boundary it moves back to the previous geodesic and then moves again to the next geodesic on the side that was not exhausted. The total distance traversed is the same except for the last step in which is increased by $2\,(d_0c^k - d_0c^{k-1})$. The competitive ratio is given by the distance traversed divided over the length of a shortest recognition path. As we have seen above the length of a shortest recognition path at step k is at least d_0c^{k-1}. Therefore,

$$\mathcal{C} = \min_{c>1} \max_{k\in N} \left\{ \frac{d_0 c^k + 2\pi d_0 (c^{k+1} - 1)/(c-1) + 2\left(d_0 c^k - d_0 c^{k-1}\right)}{d_0 c^{k-1}} \right\}$$
$$= \min_{c>1} \left\{ 2\pi \frac{c^2}{c-1} + 3\,(c-1) \right\}$$

Differentiation yields a minimum for $c = 1+\sqrt{2\pi/(2\pi+3)}$, with a competitive ratio of $4\pi+1+2\sqrt{2\pi(3+2\pi)} < 28.85$. If the pocket edges were not exhausted we obtain a lower competitive ratio for the same value of c, namely ~ 27.2. Thus, the worst case competitive ration is less than 28.85. □

6 Conclusions

We have presented a strategy for on-line searching in a star-shaped polygon and for on-line recognition of a star-shaped polygon. Our strategies have constant competitive ratios of 11.52 and 28.85 respectively, independently of the starting position of the robot and the position of the target. This is in contrast to on-line searching in other classes of polygons where both the position of the target and the starting position are heavily limited.

We have also presented a lower bound for on-line searching in a star-shaped polygon which is close to the upper bound obtained by our strategy. We show that no strategy which walks into the kernel of a star-shaped polygon can do better than 1.50 which improves on the best previously known lower bound of $\sqrt{2}$ [8]. Finally, we show that recognition of a star-shaped polygon is at least $\sqrt{82}$ competitive.

References

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

[2] P. Berman, A. Blum, A. Fiat, H. Karloff, A. Rosén and M. Sacks. “Randomized robot navigation algorithms”, *Proceedings of 7th ACM-SIAM Symposium on Discrete Algorithms*, (1996).

[3] A. Blum, P. Raghavan and B. Schieber. “Navigating in unfamiliar geometric terrain”, *Proceedings of 23rd ACM Symposium on Theory of Computing*, (1991), pp. 494-504.

[4] Ch. Bröcker and A. López-Ortiz . “Position-independent street searching”, In F. Dehne, A. Gupta, J-R. Sack and R. Tamassia, editors, *Proc. 6th Internationa*

Workshop on Algorithms and Data Structures, LNCS 1663, pages 241–252. Springer Verlag, 1999y.

[5] 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.

[6] A. Datta and Ch. Icking. “Competitive searching in a generalized street”, *Proceedings 10th ACM Symposium on Computational Geometry*, (1994), pp. 175-182.

[7] S. Gal. *Search Games*, Academic Press, 1980.

[8] Ch. Icking and R. Klein. “Searching for the kernel of a polygon. A competitive strategy”, *Proceedings 11th ACM Symposium on Computational Geometry*, (1995).

[9] Ch. Icking, R. Klein and E. Langetepe. “Searching for the kernel of a polygon: a competitive strategy using self-approaching curves”. Technical Report 211, Department of Computer Science, FernUniversität Hagen, Germany, 1997.

[10] Ch. Icking, R. Klein and E. Langetepe. “An optimal competitive strategy for walking in streets.” In *Proc. 16th Symp. on Theoretical Aspects of Computer Science*, 1999.

[11] Ch. Icking, R. Klein and Lihong. Ma. “How to look around a corner.” In *Proc. 5th Canadian Conference on Computational Geometry*, pp. 443–448, 1993.

[12] M.-Y. Kao, J. H. Reif and S. R. Tate. “Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem”, *Proceedings of 4th ACM-SIAM Symposium on Discrete Algorithms*, (1993), pp. 441-447.

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

[14] J. Kleinberg. “On-line search in a simple polygon”, *Proceedings of 5th ACM-SIAM Symposium on Discrete Algorithms*, (1994), pp. 8-15.

[15] D.T. Lee and F.P. Preparata. “An optimal algorithm for finding the kernel of a polygon”, *Journal of the ACM*, Vol. **26**, (1979), pp. 415-421.

[16] J-H. Lee, K-Y. Chwa. “Tight analysis of a self-approaching strategy for the online kernel-search problem”, *Inf. Proc. Let.*, Vol. 69, No. 1, 1999, pp.39-45.

[17] J-H. Lee, C-S. Shin, J.H. Kim and S.Y. Shin. “New competitive strategies for searching in unknown star-shaped polygons”, *Proc. 13th ACM Symp. on Computational Geometry*, (1997).

[18] A. López-Ortiz and S. Schuierer. “Generalized streets revisited”, In J. Diaz and M. Serna, editors, *Proc. 4th European Symposium on Algorithms*, LNCS 1136, pages 546–558. Springer Verlag, 1996.

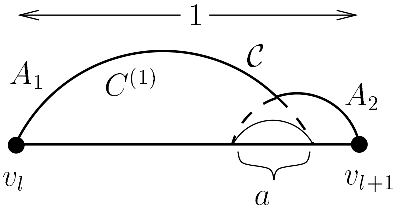

Fig. A.1. The situation for two circles.

[19] A. López-Ortiz and S. Schuierer, "Position-independent near optimal searching and on-line recognition in star polygons", *Proc. 5th Workshop on Algorithms and Data Structures (WADS)*, (1997), Lecture Notes in Computer Science, pp. 284-296.

[20] A. López-Ortiz and S. Schuierer. "The ultimate strategy to search on m rays?" In W.-L. Hsu and M.-Y. Kao, editors, *Proc. 4th Intl. Conf. on Computing and Combinatorics*, volume 1449 of *LNCS*, pages 75–84, 1998.

[21] L. Palios, "A new competitive strategy for reaching the kernel of an unknown polygon", To appear, *Proc. 7th SWAT*, 2000.

[22] F. P. Preparata, M. I. Shamos. *Computational Geometry,* Springer-Verlag, New York, 1985.

[23] S. Schuierer and I. Semrau. "An optimal strategy for searching in unknown streets." In *Proc. 16th Symp. on Theoretical Aspects of Computer Science.* to appear, 1999.

A Computing the length of $\mathcal{C}_l$

Lemma 4 *The length of* $\mathcal{C}_l$ *is at most* $(\pi - \theta)/\sin\theta\ d(v_l, v_{l+1})$. [3]

Proof: The proof is by induction. First note that we can replace all parts of edges $C^{(j)}$ that are part of $\mathcal{C}_l$ by circle segments that intersect the end points of $C^{(j)}$. Hence, we assume in the following that $\mathcal{C}_l$ consists only of circle segments.

The argument clearly holds if $\mathcal{C}_l$ consists of only one circular arc. So assume that the claim is true for all curves that consist of k circular arcs and consider a curve that consists of $k+1$ circular arcs that belong to the circles $C^{(1)}, \ldots, C^{(k+1)}$ from left to right. Note that $\mathcal{C}_l$ is the upper envelope of $C^{(1)}, \ldots, C^{(k+1)}$. Consider the upper envelope $\mathcal{C}_k$ of $C^{(1)}, \ldots, C^{(k)}$ and the arc A_{k+1} of $C^{(k+1)}$ that belongs to $\mathcal{C}$. Let B_k be the arc of $C^{(k)}$ that is contained in $C^{(k+1)}$. Clearly, the length of $\mathcal{C}$ is given by the length of $\mathcal{C}_k$ minus the length of B_k plus the length of A_{k+1}. Finally, let D_j be the length of the part of $\overline{v_l v_{l+1}}$

[3] For the definition of $\mathcal{C}_l$ refer to page 8.

that is contained in the disks spanned by $C^{(1)}, \dots, C^{(j)}$, $1 \leq j \leq k+1$. We are interested in the ratio

$$r_{k+1} = \frac{|\mathcal{C}|}{D_{k+1}} = \frac{|\mathcal{C}_k| - |B_k| + |A_{k+1}|}{D_{k+1}}.$$

By the induction hypothesis $|\mathcal{C}_k| \leq (\pi - \theta)/\sin\theta \; D_k$. If we write $D_{k+1} = D_k + \Delta D$ where ΔD is the increase in the diameter of $\mathcal{C}$ caused by $C^{(k+1)}$, then

$$r_{k+1} \leq \frac{(\pi - \theta)/\sin\theta \; D_k - |B_k| + |A_{k+1}|}{D_k + \Delta D}.$$

If $|A_{k+1}| - |B_k| < (\pi - \theta)/\sin\theta \; \Delta D$, then the claim follows immediately. Hence, we can assume that $|A_{k+1}| - |B_k| \geq (\pi - \theta)/\sin\theta \; \Delta D$ which implies that the above ratio decreases monotonically in D_k. Hence, D_k should be chosen as small as possible. The smallest value for D_k is the line segment contained in $C^{(k)}$ which implies that $\mathcal{C}_k$ consists only of $C^{(k)}$ and all other circles have radius 0. Hence, we have to check if the ratio for two circles is less than or equal to $(\pi - \theta)/\sin\theta$ and the claim follows.

Consider the situation in Figure A.1. Let $\mathcal{C}$ be the upper hull of the two circles $C^{(1)}$ and $C^{(2)}$ and A_i the part of $C^{(i)}$ that is contained in $\mathcal{C}$, for $i = 1, 2$. W.l.o.g. we can assume that the distance from the left end point to the right end point of $\mathcal{C}$ is one. We denote the length of the overlap of the two circle segments by a. We want to show that the sum of the lengths of A_1 and A_2 is no more than $(\pi - \theta)/\sin\theta$. To see this consider the sum of the lengths of the complete upper circle segments of $C^{(1)}$ and $C^{(2)}$ which equals $(1 + a)(\pi - \theta)/\sin\theta$. In order to obtain the length of A_1 and A_2 we need to substract the length of the dashed circle segments in Figure A.1. The sum of the length of these segments is clearly larger than the length of the circle segment over the line segment of length a. Therefore,

$$|A_1| + |A_2| \leq |C^{(1)}| + |C^{(2)}| - a\frac{\pi - \theta}{\sin\theta} = (1 + a)\frac{\pi - \theta}{\sin\theta} - a\frac{\pi - \theta}{\sin\theta} = \frac{\pi - \theta}{\sin\theta}$$

as claimed. This concludes the proof that the length of $\mathcal{C}$ is at most $(\pi - \theta)/\sin\theta$. □

B Walking into the kernel

Consider a coordinate system as shown in Figure B.1. The starting position of the robot has coordinates $(1, -1)$. The possible kernel locations are, in the

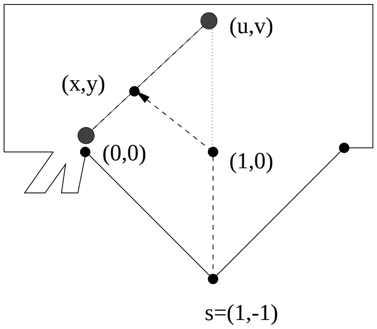

Fig. B.1. Walking into the kernel.

limit, at positions $(0,0)$ and (u,v) where (u,v) is a point on a line ℓ at an angle α to the horizontal.

The optimal robot path consists of (1) a path from the start point $(1,-1)$ to $(1,0)$, (2) a path from $(1,0)$ to a point (x,y) within a distance of $5/2\,\epsilon$ or less from ℓ and (3) a path from (x,y) to the kernel selected by the adversary. As $\epsilon \to 0$, (x,y) becomes arbitrarily close to ℓ.

The distance traversed by the robot is at least the length of the polygonal chain through the start/end points of steps (1)-(3) above. Since (x,y) and (u,v) are in ℓ and $u=1$, we have that $y = x\,\tan\alpha$ and $(u,v)=(1,\tan\alpha)$. Thus the robot's path has length

$$\begin{aligned}|\Gamma_0| &\geq d((1,-1),(1,0)) + d((1,0),(x,y)) + d((x,y),(0,0)) \\ &= 1 + \sqrt{(1-x)^2 + (x\tan\alpha)^2} + \frac{x}{\cos\alpha}\end{aligned}$$

if the kernel is located at $(0,0)$ and

$$\begin{aligned}|\Gamma_u| &\geq d((1,-1),(1,0)) + d((1,0),(x,y)) + d((x,y),(u,v)) \\ &= 1 + \sqrt{(1-x)^2 + (x\tan\alpha)^2} + \frac{1-x}{\cos\alpha}\end{aligned}$$

if the kernel is located at (u,v). The competitive ratio of the strategy is given by

$$\max\left\{\frac{|\Gamma_u|}{d((1,-1),(u,v))}, \frac{|\Gamma_0|}{\sqrt(2)}\right\}$$

The expresion above is minimized when the two terms are equal. Substituting the angle $\alpha = 655/1000 = 0.655$ selected by the adversary in the equality

$$\frac{|\Gamma_u|}{d((1,-1),(u,v))} = \frac{|\Gamma_0|}{\sqrt(2)}$$

results in a quadratic expression. The exact solution can readily be computed using a symbolic algebra package such as Maple. The closed form solution for x has 42 terms, which we omit for obvious reasons. Substituting the x value into either of terms say, $|\Gamma_0|/\sqrt(2)$ gives a closed form for the competitive ratio which evaluates numerically to 1.492.