Position-Independent Near Optimal Searching and On-line Recognition in Star Polygons*

Alejandro López-Ortiz[1] and Sven Schuierer[2]

[1] Research Scientist, Open Text Corp., 180 Columbia St. W., Waterloo, Ontario CANADA N2L 5Z5, e-mail: alopez-o@daisy.UWaterloo.ca

[2] Institut für Informatik, Universität Freiburg, Am Flughafen 17, Geb. 051, D-79110 Freiburg, FRG, e-mail: schuiere@informatik.uni-freiburg.de

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 polygon with a competitive ratio of 14.5, and we further refine it to 12.72. This makes star polygons the first non-trivial class of polygons known to admit constant competitive searches independent of the position of the target. We also provide a lower bound of 9 for the competitive ratio of searching in a star polygon—which is close to the upper bound.

A similar task consists of the problem of on-line recognition of star polygons for which we also present a strategy with a constant competitive ratio *including* negative instances.

1 Introduction

In the past years on-line searching has been an active area of research in Computer Science (e.g. [1, 2, 4, 7, 8, 11]). In its full generality, an on-line search problem consists of an agent or robot searching for a target on an unknown terrain. In the worst case a search by a robot on 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.

The robot is assumed to be equipped, as it is standard in the field, 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. The performance of an on-line search strategy is 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 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 known classes of polygons that admit search strategies for some targets with a constant competitive ratio, most notably *streets* [7], *$\mathcal{G}$-streets* [4, 10], *HV-streets* [3] and *θ-streets* [3]. However, the existence of a constant competitive searching strategy for these classes of polygons is strongly dependent on the position of the target.

A natural question is to find a class of polygons which the robot may search at a constant competitive ratio independently of the position of the target. Since the target might be hiding anywhere inside the polygon, a natural choice is to explore the class of polygons where one polygon can be seen in its entirety from a single point, known as star polygons.

Icking and Klein studied the problem of on-line *kernel* searching in a star polygon. In this case, the competitive ratio is given by the ratio of the length traversed by the robot from the starting point to a kernel point and the optimal distance, which is the

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

the distance from the starting point to the kernel set. In [4] Icking and Klein presented a ~ 5.81 competitive strategy for walking into the kernel of a star polygon.

In this paper we present the first non-trivial class of polygons which admits constant competitive-ratio position-independent target searching. In section 2 we introduce some concepts and definitions of use in searching polygons. In section 3 we present a 14.5-competitive algorithm for target searching in star polygons and prove a lower bound of 9 for the competitive ratio of any search strategy for star polygons, we further refine this strategy to achieve a competitive-ratio of 12.72. In section 4 we use this strategy to construct the first constant competitive algorithm for recognition of star polygons. That is, given a polygon, the robot follows a path that proves or disproves that the polygon is a star where the path is no more than a constant times longer than a shortest path with the same property. Furthermore, such path leads into the kernel in a constant competitive ratio as well. We also improve the $\sqrt{2}$ lower bound for walking into the kernel of a star polygon to ~ 1.48.

2 Definitions

We say two points p_1 and p_2 in a polygon P are visible to each other if the line segment $\overline{p_1 p_2}$ is contained in P.

Definition 1. Let p be a point in P. The visibility polygon of p is the subset of P visible to p and denoted by $V_P(p)$.

We assume that the robot has access to its local visibility polygon by a range sensing device, e.g. a ladar.

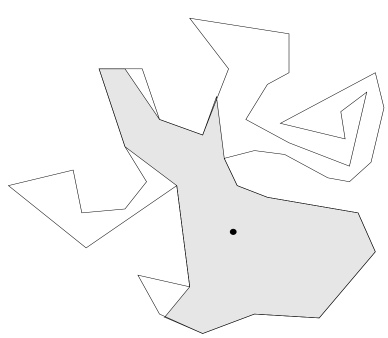

Fig. 1. Visibility polygon.

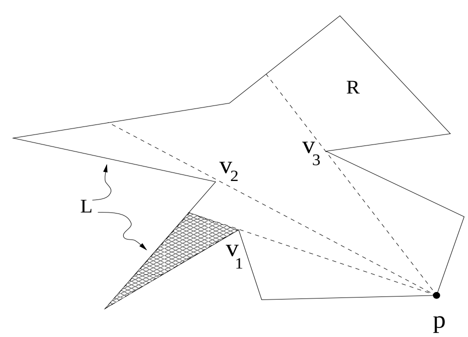

Fig. 2. Left and right pockets.

Definition 2. [12] A simple polygon P is a star polygon if there exists a point z in P such that $V_P(z) = P$. The set of all points z inside P with $V_P(z) = P$ is the kernel of P.

Star polygons are often referred to as star-*shaped* polygons [12], in this paper we use the equally common but shorter name of star polygons.

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 consists of some polygon edges and a single line segment not belonging to the boundary of P. The edge of the pocket which is not a polygon edge is called the window of the pocket. Note that a window intersects the boundary of P only in its end points. More generally, a line segment that intersects the boundary of P only in its end points is called a chord.

A pocket edge of p is a ray emanating from p which contains a window. Each pocket edge passes through at least one reflex vertex of the polygon, which is also an end point of the window associated with the pocket edge. This reflex vertex is called the entrance point of the pocket.

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. Right pocket and right pocket edge 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 to the right of all left pocket edges and 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 $\overrightarrow{pv_1}$ and $\overrightarrow{pv_2}$ and to the left of $\overrightarrow{pv_3}$.

This also implies that, for star polygons, starting from a left pocket and moving clockwise, all left pocket edges appear consecutively; at some point the first right pocket edge is seen and from then onwards all pocket edges are right pocket edges, until the full circle back to the sequence of left pocket edges is completed. This is so as the extension of each pocket defines a half plane which contains the kernel of P, if the pockets were to alternate between left and right, the intersection of these halfplanes would be empty which is a contradiction.

If the robot is initially located on a point s on the boundary of the polygon, the robot can scan all left pocket edges by starting from the edge on which s lies, and proceeding on the clockwise direction the interior of the polygon. At some point, a right pocket edge is seen and from then onwards all pocket edges are right pocket edges until the robot reaches the edge containing s again, which completes the scanning process.

3 Target Searching in Star Polygons

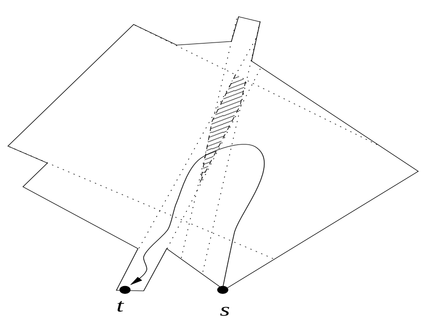

Fig. 3. Searching for a target via the kernel.

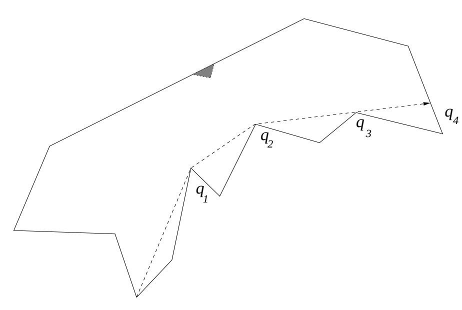

Fig. 4. An extended pocket edge.

There are many similarities between searching for the target and searching for the kernel. However, note that in general, when searching for a target, it is not an efficient strategy to first go to the kernel or towards the center and from there move to the target as illustrated in Figure 3. As illustrated in this case, a path advancing towards the kernel can be made arbitrarily larger than the distance from s to t.

Searching for a target of unknown location inside a star polygon is a provably harder problem than searching for the kernel, as we shall see in the second part of this section.

First we present a strategy to search for a target in a star polygon.

Consider the set of pocket edges seen by the robot from the starting position. We extend this set as follows.

Definition 3. Given a polygon P, an extended pocket edge from a point s is a polygonal chain $q_0, q_1, q_2, \ldots, q_k$ such that $q_0 = s$, and each of q_i is a reflex vertex of P, save possibly for q_k. Furthermore q_{k-2}, q_{k-1} and q_k are collinear and form a pocket edge with $\overline{q_{k-1}q_k}$ as associated window. If $\overline{q_{k-2}q_k}$ is a left (right) pocket edge, then each of $\angle q_{i-1}q_iq_{i+1}$ is a counterclockwise (clockwise) reflex angle (see Figure 4).

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.

Lemma 4. *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. 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. □

Theorem 5. *There exists a strategy for searching for a target inside a star polygon with a competitive ratio of at most 14.5.*

Proof. Let $\mathcal{F}$ denote the set of all extended pocket edges starting from s. From the definition it follows that, in general, the robot may not see all of $\mathcal{F}$ from s (see for example the star polygon of Figure 5). The robot thus uses a strategy that starts with a subset $\mathcal{F}_0$ of $\mathcal{F}$. This set is enlarged as the robot sees new pocket edges. Given an extended pocket edge E, let l_E denote the last point in the chain, and p_E denote the second to last point of E.

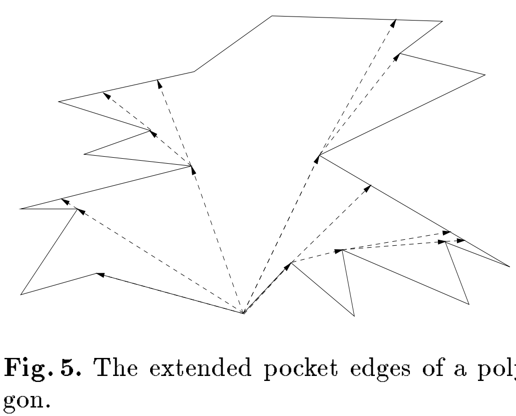

Fig. 5. The extended pocket edges of a polygon.

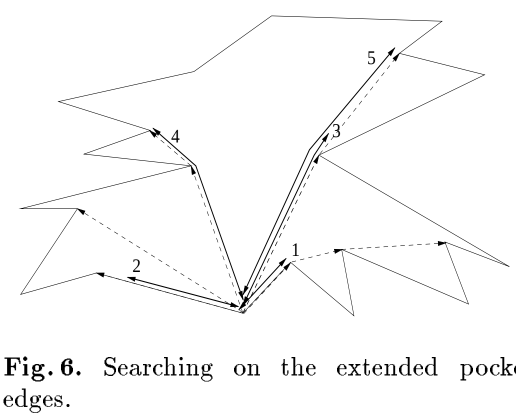

Fig. 6. Searching on the extended pocket edges.

Let *side* $\in$ {*left*, *right*} and if *side* $= right$, then $\neg$*side* $=$ *left* and vice versa.

Algorithm Star Search
Input: A star polygon P and a starting point s;
Output: The location of the target point t;

```
1   let ℱ be the set of extended pocket edges currently seen but not explored;
    (* Initially ℱ contains only simple pocket edges; *)
2   let p_E be the closest entrance point to s and d = d(p_E, s)
3   if E is a left pocket edge then let side ← left
4     else let side ← right;
5   while ℱ is non-empty do
6      traverse d units on E starting from s;
7      if t is seen then exit;
8      add the new pocket edges seen in this trajectory to ℱ as extended pocket
           edges starting from s;
9      remove from ℱ all extended side pocket edges to the side side of the extended
           pocket edge \overline{sp_E}, including E if p_E is reached;
10     move back to s;
11     side ← ¬side; d ← c · d;
12     if side = left
13       then let p_E be the rightmost entrance point on a left pocket edge such
                that the length of the extended pocket edge from s to p_E is less than
                d
14              if there is no such edge
15                then let E be the leftmost edge in ℱ
16     if side = right
17       then select E analogously to the case side = left;
    end while;
```

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

Note that after the first two iterations the while-loop has the following invariant:

> *Invariant*: All pockets at a distance of d/c^2 or less on the *side* side have been explored.

The correctness of the algorithm follows from Observation 1 and Lemma 4 as follows. As the robot visits extended pocket edges, it eventually visits the leftmost right pocket edge and the rightmost left pocket edge if t is not found before.

Once the robot has visited the extreme leftmost and rightmost pocket edges, it has explored the part to the left of the extreme left-pocket edge, and to the right of the extreme right-pocket edge. Furthermore, the part of the polygon contained in between the two extreme pocket edges has no hidden regions as it contains no pockets. Thus, the entire polygon is seen, and the target must have been found.

We claim that Algorithm *Star Search* has a competitive ratio of 14.5. At the end of Step 17, the invariant holds because if there was a, say, left pocket at a distance of less than d/c^2 it means it was part of the set $\mathcal{F}$ two steps before. Thus, if it was unexplored then, it either was traversed, or another left pocket of length at most d/c^2 which is to the right of it was traversed. But exploring this second edge entails exploring the earlier edge as shown in Lemma 4.

A consequence of the invariant is that if the current distance to be traversed by the robot is d, then the target cannot be at a distance of less than d/c^2. The worst case occurs when the robot sees the target at a distance of $d/c^2 + \epsilon$, at the very end of a search of length d (see Figure 7). This means that the ratio of the distance traversed by the robot according to Algorithm *star search* to the distance from s to t is at most

$$2\frac{\sum_{i=0}^{n} c^i}{c^{n-2}} + 1 = \frac{2c^3}{c-1} + 1 - O(1/c^{n-1}).$$

Substituting the value 3/2 which minimizes $2c^3/(c-1)$ gives a competitive ratio of $1 + 27/2 = 14.5$. In fact, it can be shown that there is no choice of the step lengths that yields a better competitive ratio for the above algorithm [1, 5]. □

We observe that the worst case configuration occurs when the angle $\angle L_{i-2}sL_i$ is relatively flat. In this case the competitive ratio can be improved if the robot does not follow the straight line segment $\overline{sL_i}$ but follows a curve that allows it to detect the target earlier (see Figure 7).

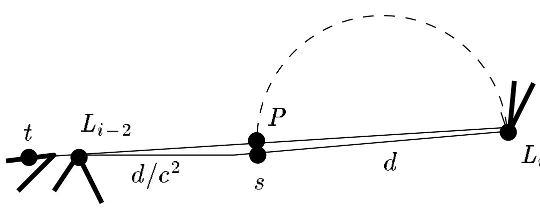

Fig. 7. The worst case to discover the target. If the robot follows the dashed path, then t is detected at P instead of L_i.

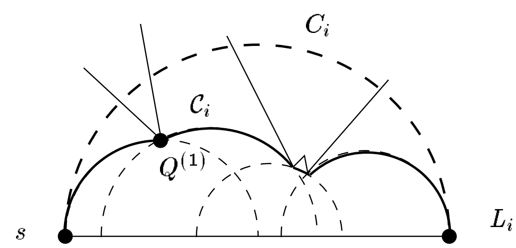

Fig. 8. The new strategy of the robot.

So instead of traveling along the line segment $\overline{sL_i}$ the robot now travels along the semi-circle C_i that is spanned by $\overline{sL_i}$. More precisely, the robot computes a curve $\mathcal{C}_i$ that connects s and L_i and that consists of parts of circles $C^{(1)}, \ldots, C^{(k_i)}$ as follows. The center $c^{(j)}$ of each circle $C^{(j)}$ is contained in $\overline{sL_i}$ with $c^{(j)}$ to the left of $c^{(j+1)}$, for $1 \le j \le k_i - 1$. The curve $\mathcal{C}_i$ is defined inductively. The circle $C^{(1)}$ is the first circle with its center to the right of s that contains s and intersects either L_i or the boundary of P in a point $Q^{(1)}$ above $\overline{sL_i}$. The part of $C^{(1)}$ between s 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 $1 \le j \le k_i - 1$. There is a point $Q^{(j)}$ such that $C^{(j)}$ intersects the boundary of P in $Q^{(j)}$. The circle $C^{(j+1)}$ is the first circle with its center to the right of $c^{(j)}$ that contains $Q^{(j)}$ and intersects either L_i or the boundary of P in a point $Q^{(j+1)}$ different from $Q^{(j)}$ above $\overline{sL_i}$. For illustration refer to Figure 8. Because of the limited space and the fairly involved analysis of the above strategy, we just mention that the competitive ratio can be improved to 12.72 in this way.

3.1 A Lower Bound on the Competitive Ratio

In this section we prove a lower bound of nine for the competitive ratio for searching in star polygons. Our proof is based on the following theorem about on-line searching on the line.

Theorem 6. [9, **Theorem 2.2**] *Any on-line search strategy on the line for a target at a distance of at most D is at least $(9 - f(D))$-competitive, where $f(n) \le 24/\log_4 n$, for sufficiently large D.*

Theorem 7. *Any strategy for searching for a target inside a star polygon is at least 9-competitive.*

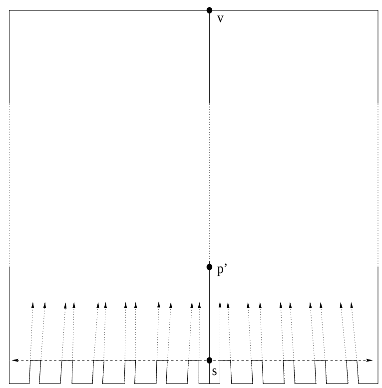

Fig. 9. Lower bound for searching for a target.

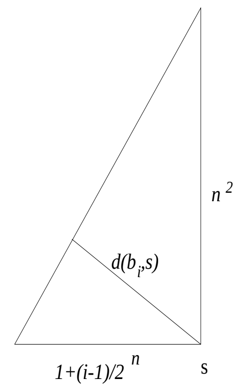

Fig. 10. Distance to a beam.

Proof. Consider the polygon of Figure 9. Let s be located at the origin. This polygon is made of $(n-1)2^{n-1}+1$ teeth with $(n-1)/2^{n-3}+4$ vertices attached to a rectangle of height n^2 and width $2n$. Teeth are equally spaced at a distance $1/2^n$, and of width $1/2^{n+1}$ save for the tooth containing s which is of width $2-1/2^n$. Each tooth defines a beam (see Definition 9). All beams intersect at the point $v=(0,n^2)$ which sees the entire interior of the polygon.

We claim that the robot must essentially do a doubling search on the teeth, in which case Theorem 6 gives a lower bound of 9. However, in this case there are several differences that must be considered. First, the movement of the robot is not restricted to a line; second, the lower bound is for searches on any point of the interval rather than on discrete positions. Thus the proof proceeds as follows: first we argue that any search strategy is sufficiently close to a search on the real line, and secondly we show that the bound for the continuous case implies a similar bound for the discrete case.

For the robot to explore a tooth it must reach the beam above it. We number the beams symmetrically, and consecutively starting from the origin; thus beam b_i is at the same distance from the origin as beam $-b_i$. The distance from s to the the base of the ith beam on either side is $d_i=1+(i-1)/2^n$. The distance from s to the closest point in the beam is (see Figure 10)

$$d(b_i,s)=\frac{d_i}{\sqrt{1+(1+(d_i)^2/n^4}}\geq\frac{d_i}{\sqrt{1+1/n^2}}$$

However the robot is not forced to move back to s after each search. Since the robot cannot reach a height past $9n$ as that alone would imply a competitive ratio above 9 we consider the point p' located at $(0,9n)$ and it follows that

$$d(b_i,p')\geq d_i\frac{n^2-9n}{\sqrt{n^4+d_i^2}}\geq d_i\frac{n^2-9n}{\sqrt{n^4+n^2}}=d_i\frac{n-9}{\sqrt{n^2+1}}$$

The order in which beams are visited can be denoted by the sequence $S=\{s_j\}_{1\leq j\leq N}$ of the distances d_i from the origin to the base of those beams in which the robot changed direction (turn points).

Consider the beams associated to two consecutive terms in the sequence S above, say b_{k_i} and $b_{k_{i+1}}$. Without loss of generality, let us assume that b_{k_i} is on the left side and $b_{k_{i+1}}$ on the right side. Then, the distance traversed by the robot from beam b_{k_i} to beam $b_{k_{i+1}}$ is at least $d(q_i,q_{i+1})\geq d(q_i,p_i)+d(p_i,q_{i+1})$, where q_j denotes the position of the robot in b_{k_j} for $j = \{i, i+1\}$, and p_i is the intersection of $\overline{q_i q_{i+1}}$ with the y-axis. Furthermore, $d(q_j, p_j) \geq d(b_{k_j}, p_j) \geq d(b_{k_j}, p')$.

Let $\mathcal{C}_S$ denote the competitive strategy of the strategy S on the real line. Analogously, let $\mathcal{C}_S^D$ denote the competitive ratio of a strategy S in the discrete case.

Now we will show that the search strategy S, applied to a target hiding at a point at distance d_i on the real line has competitive ratio:

$$\mathcal{C}_S^D \geq \sup_{1 \leq j \leq N} \left\{ 1 + 2 \frac{\sum_{i=1}^{j} |s_i|}{|s_{j-1}| + 1/2^n} \right\} \geq 9 - 25/\log_4 n.$$

Assume that, to the contrary, $\mathcal{C}_S^D < 9 - 25/\log_4 n$. We know from Theorem 7 that any sequence S visiting the interval $[-n, n]$ and searching for a target located in any interior point has a competitive ratio greater or equal to $9 - 24/\log_4 n$. Since S is such a sequence, we have then that

$$\mathcal{C}_S = 1 + 2 \frac{\sum_{i=1}^{k} |s_i|}{|s_{k-1}|} \geq 9 - 24/\log_4 n$$

for some k such that $1 \leq k \leq N$. Let $C_S(s_k)$ and $C_S^D(s_k)$ denote the competitive ratio of strategy S to find a target hiding at the point s_k for the real and discrete case respectively. Note that $\mathcal{C}_S = C_S(s_k) \geq 9 - 24/\log_4 n$ and that

$$1 + 2 \frac{\sum_{i=1}^{k} |s_i|}{|s_{k-1}| + 1/2^n} \leq C_S^D(s_k) < 9 - \frac{25}{\log_4 n} \quad \Longrightarrow \quad \sum_{i=1}^{k} |s_i| < \left(8 - \frac{25}{\log_4 n}\right) \left(|s_{k-1}| + \frac{1}{2^n}\right).$$

The additive factor of $1/2^n$ in the denominator accounts for the next possible position of the target on that side. We claim that $0 \leq C_S(s_k) - C_S^D(s_k) < 1/\log_4 n$. Indeed,

$$\begin{aligned} C_S(s_k) - C_S^D(s_k) &= 2 \sum_{i=1}^{k} |s_i| \left[\frac{1}{|s_{k-1}|} - \frac{1}{|s_{k-1}| + 1/2^n} \right] \\ &< \left(8 - \frac{25}{\log_4 n}\right) \frac{1}{2^{n-1} |s_{k-1}|} \leq \frac{1}{2^{n-4}} \leq \frac{1}{\log_4 n} \qquad \text{for } n > 3 \end{aligned}$$

as claimed. Thus, from $C_S(s_k) \geq 9 - 24/\log_4 n$ it follows that $\mathcal{C}_S^D \geq C_S^D(s_k) > 9 - 25/\log_4 n$, which is a contradiction.

Thus it follows that $\mathcal{C}_S^D \geq 9 - 25/\log_4 n$. Now, we know that the robot traversed, for each s_j a distance $d(q_j, p_j)$ which is at least $|s_j| \frac{n-9}{\sqrt{n^2+1}}$, for a total competitive ratio of at least

$$1 + \sup_{k \in Z} \left\{ \frac{n-9}{\sqrt{n^2+1}} \left(C_S^D(s_k) - 1 \right) \right\} \geq 1 + \sup_{k \in Z} \left\{ \left(\frac{n-9}{\sqrt{n^2+1}} \right) \left(8 - \frac{25}{\log_4 n} \right) \right\}$$

The value above is a lower bound for the competitive ratio of the robot searching a polygon. As the construction of the polygon of Figure 9 is valid for any n, we have that, in the limit, the competitive ratio is bounded by

$$\lim_{n \to \infty} 1 + \left(\frac{n-9}{\sqrt{n^2+1}} \right) \left(8 - \frac{25}{\log_4 n} \right) = 9$$

as claimed. $\square$

4 Recognition of Star Polygons

For the on-line star recognition problem, we assume that given a polygon, the robot aims to determine if it is star shaped. Similarly to target searches, the competitive ratio is given by the quotient between the shortest path that proves or disproves that a given polygon is a star and the distance traversed by the robot.

As Figure 11 shows, the problem of on-line search for the kernel of a polygon is at least $\sqrt{2}$ competitive [6].

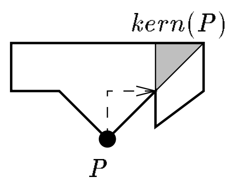

Fig. 11. A lower bound of $\sqrt{2}$ for the competitive ratio of searching for the kernel of a star polygon. Any on-line strategy with a competitive ratio of $\leq \sqrt{2}$ has to follow the dashed path.

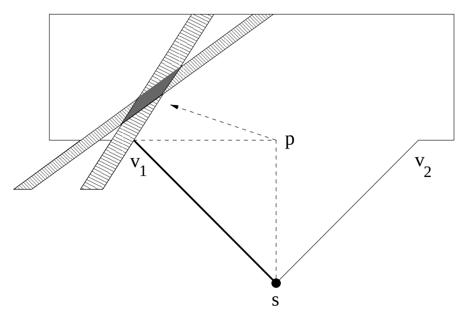

Fig. 12. Polygon with two beams.

The next theorem shows that kernel searches are strictly worse than $\sqrt{2}$-competitive. This result stands out against several other lower bounds for searching in simple domains, for which it seems that a robot can find an optimal path on-line for the L_1 metric [8].

Definition 8. 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 9. Given the current position of the robot p and a pocket B with respect to that point, 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 12).

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

Theorem 10. *Searching for the kernel of a polygon is at least* $1/2 + 3/8\sqrt{2} + (2 + \sqrt{2})/8\sqrt{10\sqrt{2} - 13} \sim 1.48$*-competitive.*

Proof. Consider the polygon of Figure 12. Notice that the robot must reach the line segment $\overline{v_1 v_2}$ before it reaches the kernel. As well, the robot must reach $\overline{v_1 v_2}$ at its midpoint p, as otherwise the following construction can be made on the opposite side and it follows from the triangle inequality that the competitive ratio would only worsen. Again, 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 13, such that one is next to v_1 the other right above the midpoint, and the line joining them is at a $\pi/4$ angle to the horizontal. This can be achieved by locating a beam A along the line

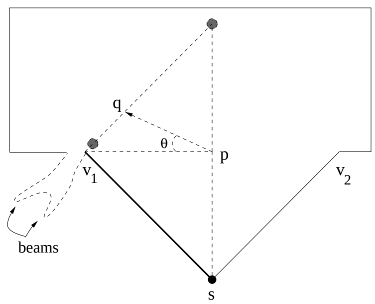

Fig. 13. Lower bound configuration.

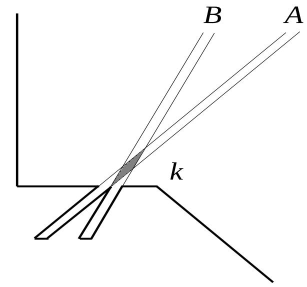

Fig. 14. Progressively thinner beams.

joining the two candidate regions, and a second one, B, nearly parallel and to the right of A (see Figure 14). The intersection of both beams defines the kernel of visibility.

At this point, 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.

In this case, the robot cannot decide which of the candidates is the kernel before it reaches at least one of A or B. As the beams become progressively thinner, the robot reaches either beam at an ϵ distance of the $\pi/4$ line joining the two candidate regions (that is, the right edge of the A beam).

Assume this happens at a point q located, as indicated in the previous paragraph, arbitrarily close to the $\pi/4$ line. Let θ be the angle given by $\angle v_1 pq$. Without loss of generality, let the distance $d(s, v_1) = \sqrt{2}$. We compute the competitive ratio on the left. Let C_1 and C_2 be the two candidate regions. To compute the competitive ratio we first notice that $d(C_1, C_2) = \sqrt{2} - \epsilon$. Then from elementary trigonometry we obtain $d(q, C_1) = d(p, q)\sin(\theta)/\sin(\pi/4)$ and $1 - d(q, C-2) = d(p, q)\cos(\theta)$, from which follows

$$d(p, q) = \frac{\sin(\pi/4)}{\sin(\pi/4 + \theta)} \text{ and } d(q, C_1) = \frac{\sin(\theta)}{\sin(\theta + \pi/4)}.$$

Similarly, $d(q, C_2) = \sin(\pi/2 - \theta)/\sin(\pi/2 - \theta + \pi/4)$. Thus the competitive ratio for the kernel on the left side is given by

$$\frac{\sqrt{2}\sin(\theta) + \sqrt{2}\cos(\theta) + \sqrt{2} + 2\,\sin(\theta)}{2\,\sin(\theta) + 2\,\cos(\theta)}$$

and on the right side

$$\frac{\sin(\theta) + \cos(\theta) + 1 + \sqrt{2}\cos(\theta)}{2\,\sin(\theta) + 2\,\cos(\theta)}$$

As the competitive ratio is the maximum of both quantities above, the robot selects θ such that the competitive ratio on either side is the same. Solving the equation we obtain,

$$\theta = \arctan\left(1/4 + 1/8\sqrt{2}\left(1 - \sqrt{10\sqrt{2} - 13}\right)\right).$$

For this value, the competitive ratio is

$$(2 + \sqrt{2})/8\sqrt{10\sqrt{2} - 13} + 1/2 + 3/8\sqrt{2} \approx 1.48642$$

as required. □

The best known search strategy for finding the kernel of a given star polygon, is by Icking and Klein [6] and results in a no worse than $\sqrt{4 + (2+\pi)^2} \sim 5.5$-competitive strategy. However, it is unclear if the same algorithm applied to a general polygon would terminate at a constant competitive ratio for negative instances. A modification of the target searching strategy of Theorem 5 can be used for this purpose. Furthermore, if the polygon is a star the proposed modified strategy reaches the kernel, if it exists, at a constant competitive ratio as well.

Definition 11. Let s be the starting position of the robot inside a polygon P. Let $V(\Gamma)$ denote the visibility region of a continuous path Γ inside P. Then we denote by OPT the length of the shortest path such that a computational agent (such as a Turing machine) can determine from $V(\Gamma)$ that P is or is not a star.

Theorem 12. *There exists a* 46.35*-competitive strategy that identifies if a polygon is or is not a star.*

Proof. The algorithm is somewhat similar to the one proposed for target searching in Theorem 5. However there are some key differences. Let *side* $\in$ {*left, right*} as before. For this theorem we say that a straight chord is a local *side* pocket edge if it joins two points which are in between two consecutive *side* extended pocket edges with one endpoint lying on the *side*-most of the two pocket edges. Similarly, local pocket edges together with the extended pocket edge in which they are anchored will be considered as extended pocket edges themselves.

The new strategy *Circle-Swipe* replaces Steps 7 and 10–11 from strategy *Star-Search* of Theorem 5.

Step 7 If the intersection of the half planes defined by the extension of the rectilinear segments of explored pocket edges becomes empty the strategy rejects. Otherwise continue until all pockets have been explored and accept.

Steps 10–11 The robot changes side $side \leftarrow \neg side$. Let $E_i = \langle q_0^i{=}s,\ q_1^i, \ldots, q_{k_i}^i \rangle$. The robot moves on a circular arc centered on q_{k-2}^i, of radius $d(q_{k-2}^i, q_{k-1}^i)$ to side *side* until it sees q_{k-3}^i. The robot then updates the radius to be $d(q_{k-3}^i, q_{k-1}^i)$ and continues describing a (new) circular arc centered at q_{k-3}^i. Eventually the robot sees s and continues describing a circle of radius d until it starts reaching the extension of edges of the next pocket edge $E_{i+1} = \langle q_0^{i+1}{=}s,\ q_1^{i+1}, \ldots, q_{k_{i+1}}^{i+1} \rangle$ to be visited on the *side* side. In each of this cases, the robot does the reverse process, reducing the radius by the length of the edge seen $(d(q_{j-1}^{i+1}, q_j^{i+1}))$ and centering the arc on q_j^{i+1} where j takes the values $0, 1, 2 \ldots k_{i+1}$ successively (see Figure 15). When the robot reaches the boundary of the polygon it returns to s moving over E_{i+1} and sets d to cd.

The invariant is now as follows.

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

Again we must show correctness and analyze the competitiveness of the strategy. In Steps 10–11, while walking on the circular arc if the robot cannot reach a *side* pocket edge, it means that the robot was blocked by the boundary of P. This boundary point must necessarily lie between the leftmost right pocket and the rightmost left pocket, as otherwise it would have been considered an extended pocket edge of the $\neg$*side* side and, chosen as target pocket edge (since it is at a distance of at most d from s).

Now if we have reached a boundary point that is to the right (left) of the rightmost (leftmost) left (right) pocket edge but before a right (left) pocket edge then this point must be visible from s. In this case the robot traverses to s and has completed searching the left (right) side and continues searching on circular arcs on the remaining side to be explored.

Since after the ith iteration of Steps 10–11 the robot has enclosed by a simple connected curve all points at distance at most dc^{i-1} it follows from Observation 4 that the invariant is correct. Step 7 accepts or rejects when either an impossibility for a star polygon has been found or the whole polygon has been explored, which is trivially correct, concluding the proof of correctness.

In turn, the analysis has two components. First we must determine the length of the worst case longest path that may be traversed up to and including the ith iteration of Steps 10–11. Secondly, we shall show that all correct algorithms must accept according to Step 7.

In iteration i the robot traverses a distance no greater than dc^i to reach the circular arc, at most $2\gamma_i dc^i$ on the arc itself for some angle γ_i and then at most dc^i back to the point s. In the worst case, in step k we surround a point at distance $dc^{k-1} + \epsilon$ as given by the invariant of Steps 10–11. Clearly, if the polygon is a star $\gamma_i \leq 2\pi$ as otherwise the intersection of the half planes determined by the edges forming and extended pocket edge would be empty.

The distance traversed before surrounding the point depends on whether or not the pocket edges on one side were exhausted. If they were, the competitive ratio is:

$$\frac{[\sum_{i=0}^{k} c^i(2\pi + 2)] + c^{k+1}(2 + \alpha) + c^{k+2}(2 + 2\pi - \alpha)}{c^k}$$
$$\leq \frac{c}{c-1}(2\pi + 2) + c(2 + \alpha) + c^2(2 + 2\pi - \alpha)$$

where $0 \leq \alpha \leq 2\pi$. Differentiation shows that the maximum is attained when $\alpha = 0$, for all c. Minimizing with respect to c we obtain a cubic equation –which can be solved symbolically– and is minimized when $c \sim 1.547$ with a competitive ratio of 46.35.

If the pocket edges were not exhausted the competitive ratio is,

$$\frac{\sum_{i=0}^{k+1} c^i(2\pi + 2)}{c^k} \leq \frac{c^2}{c-1}(2\pi + 2).$$

This ratio is smaller than 46.35 for $c = 1.547$ which gives a maximum between the two expressions of 46.35 as required.

Secondly, it is easy to see using an adversarial argument, that if the robot accepts a star polygon without having looked into all of its pockets or rejects without having found non-intersecting half planes the adversary can suitably modify the pockets and make the robot fail. That is, if the robot accepts P without exploring a pocket the adversary creates a spiral in that pocket, and the polygon is not a star. On the other hand if the robot rejects without having found non-intersecting half-planes the adversary "empties" all the pockets by means of inserting an almost flat two edge chain closing the pocket (the chain is ϵ dented by a vertex on its midpoint). Because at any time there are only a finite number of pockets and the interior of the intersection of non-degenerate set of half planes is an open set, it follows that there exists small enough ϵ such that the intersection of all the half-planes of this modified polygon is not-empty and thus the polygon is a star, contradicting the robot.

Thus we have established that an agent optimally recognizing a star-polygon traverses the shortest path Γ that satisfies the visibility conditions of step 7 for the given polygon. The invariant then states that a robot using the Circle-Swipe strategy swipes a region that is at most c times farther than the given path, for a total competitive ratio of 46.35 as computed.

□

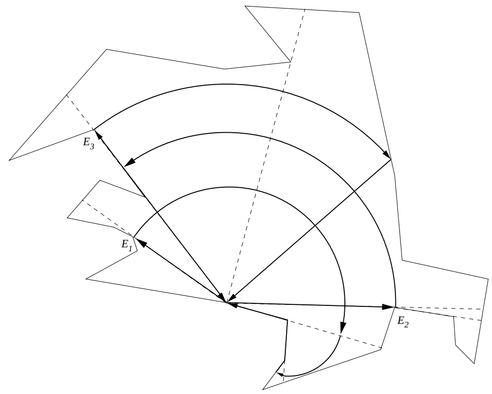

Fig. 15. Recognizing a polygon.

5 Conclusions

We have presented a strategy for on-line searching in a star polygon and for on-line recognition of a star polygon. Our strategies have constant competitive ratios independent 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 polygon which is close to the upper bound obtained by our strategy. Finally, we show that no strategy which searches for the kernel of a star polygon can achieve a competitive-ratio better than 1.48 which improves on the best previously known lower bound of $\sqrt{2}$.

References

1. R. Baeza-Yates, J. Culberson and G. Rawlins. "Searching in the plane", *Information and Computation*, Vol. **106**, (1993), pp. 234-252.
2. 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.
3. A. Datta, Ch. Hipke, and S. Schuierer. "Competitive searching in polygons—beyond generalized streets", in *Proc. Sixth Annual Int. Symp. on Algorithms and Computation*, pages 32–41. LNCS 1004, 1995.
4. A. Datta and Ch. Icking. "Competitive searching in a generalized street", *Proceedings 10th ACM Symposium on Computational Geometry*, (1994), pp. 175-182.
5. S. Gal. *Search Games*, Academic Press, 1980.
6. Ch. Icking and R. Klein. "Searching for the kernel of a polygon. A competitive strategy", *Proceedings 11th ACM Symposium on Computational Geometry*, (1995).
7. R. Klein. "Walking an unknown street with bounded detour", *Computational Geometry: Theory and Applications*, Vol. **1**, (1992), pp. 325-351.
8. J. Kleinberg. "On-line search in a simple polygon", *Proc. 5th ACM-SIAM Symp. on Discrete Algorithms*, (1994), pp. 8-15.
9. A. Lopez-Ortiz. "On-line target searching in bounded and unbounded domains", Ph.D. thesis, University of Waterloo, 1996.
10. 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.
11. A. Lopez-Ortiz and S. Schuierer. "Walking streets faster", *Proceedings of 5th Scandinavian Workshop in Algorithmic Theory Algorithms*, 1996, LNCS 1097, Springer-Verlag, pp. 345-356.
12. F. P. Preparata, M. I. Shamos. *Computational Geometry,* Springer-Verlag, New York, 1985.

This article was processed using the LaTeX macro package with LLNCS style