Position-Independent Street Searching

Christoph A. Bröcker[1] and Alejandro López-Ortiz[2]

[1] Institut für Informatik, Universität Freiburg, Am Flughafen 17, Geb. 051, D-79110 Freiburg, FRG. hipke@informatik.uni-freiburg.de

[2] Faculty of Computer Science, University of New Brunswick, Fredericton, New Brunswick, Canada, E3B 4A1. alopezo@unb.ca

Abstract. A polygon P is a street if there exist points (u, v) on the boundary such that P is weakly visible from any path from u to v. Optimal strategies have been found for on-line searching of streets provided that the starting position of the robot is $s = u$ and the target is located at $t = v$. Thus a hiding target could foil the strategy of the robot by choosing its position t in such a manner as not to realize a street.
In this paper we introduce a strategy with a constant competitive ratio to search a street polygon for a target located at an arbitrary point t on the boundary, starting at any other arbitrary point s on the boundary. We also provide lower bounds for this problem. This makes streets only the second non-trivial class of polygons (after stars) known to admit a constant-competitive-ratio strategy in the general position case.

1 Introduction

In 1991 Klein considered the problem of an agent or robot searching the interior of a simple unknown polygon for a visually identifiable target point [13]. The competitive ratio, defined as the ratio between the distance traversed by a robot and the length of the shortest path between the robot and the target, is a natural framework to evaluate the performance of a given search strategy. It is not hard to see that in general searching an arbitrary simple polygon with n vertices is $\Omega(n)$ competitive (see e.g. [13, 16]).

In the same paper, Klein introduced the class of street polygons, which can be searched on-line at a constant competitive ratio under specific restrictions on the position of the target. A polygon P is a street if there exists a pair of points (u, v) on the boundary such that the interior of the polygon is weakly visible from any path from u to v. Specifically, the strategy proposed depends on the target being located at v and the starting position of the robot being u. Several improved strategies for streets have been proposed under the same assumptions [8, 14, 16, 18]. Recently streets have been shown to be searchable at a competitive ratio of $\sqrt{2}$ in the worst case, which is optimal [20, 10], provided, as before that $s = u$ and $t = v$.

Several other classes of polygons that admit constant competitive ratios have been proposed including $\mathcal{G}$-*streets* [6, 17], *HV-streets* [5] and θ-*streets* [5]. Just as with streets, the existence of a constant competitive searching strategy for these classes of polygons is also dependent on the position of the target.

In 1997, López-Ortiz and Schuierer [15, 19] introduced the first non-trivial class of polygons known to admit a constant competitive ratio irrespective of the starting position of the robot and the target, namely star polygons.

However it remained an open question if street polygons could be searched at a constant competitive ratio when the starting position of the robot is different from u and the location of the target is not v, as all known search strategies depend heavily on this fact. In this paper we answer this question in the affirmative. This is an important generalization of the restricted street search algorithm, as otherwise a hiding target could foil the search strategy of the robot by choosing its hiding position t in such a manner as not to realize a street. This makes streets only the second non-trivial class of polygons (after stars) known to admit a constant-competitive-ratio strategy in the general case.

We propose first an algorithm for the case $s = u$ and a free target, i.e. $t \neq v$ with a competitive ratio of 36.806, and then an algorithm for the general case when $s \neq u$ with a competitive ratio of 69.216. This completes all the search cases for street polygons. It is also interesting to note that the competitive ratio can be improved significantly for rectilinear street polygons using a strategy devised for that specific case [4].

The paper is organized as follows. In Section 2 we give some basic definitions. In Section 3 we present a strategy to search a street polygon when the starting position of the robot is $s = u$ but the target is located at an arbitrary point on the boundary. In Section 4 we give a strategy for searching street polygons for arbitrary location of s and t. In Section 5 we present a lower bound of 9 for the search case when $s = u$ and of 11.78 for arbitrary position streets searching.

2 Definitions

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 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.

We say two points p_1 and p_2 in a polygon P are mutually visible if the line segment $\overline{p_1 p_2}$ 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.

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 (laser "radar").

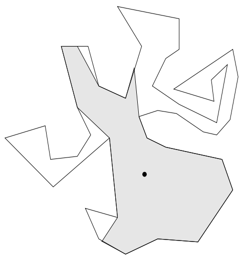

Fig. 1. Visibility polygon.

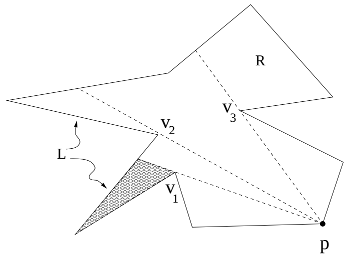

Fig. 2. Left and right pockets.

Definition 2. [**13**] *Let P be a simple polygon with two distinguished vertices, u and v, and let L and R denote the clockwise and counterclockwise, resp., oriented boundary chains leading from u to v. If L and R are mutually weakly visible, i.e. if each point of L sees at least one point of R and vice versa, then (P, u, v) is called a* street.

Streets are also known as LR-visibility polygons [3].

If the robot does not see the entire interior of P, then the regions not seen in P form connected components of $P \setminus V_P(p)$ called pockets. The boundary of a pocket is made of some polygon edges and a line segment not belonging to the boundary of P. The edge of the pocket which is not a polygon edge is called a window of $V_P(p)$. 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.

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.*

Definition 4. *We say two pocket edges p_1 and p_2 are* clockwise consecutive *if the clockwise oriented polygonal chain of $V(p)$ does not contain another pocket edge between p_1 and p_2.*

Lemma 1. *Let (P, u, v) be a street polygon. All left (right) pocket edges anchored in u are clockwise (counterclockwise) consecutive.*

It is easy to verify this by assuming otherwise and noticing then that one of the pockets cannot see the opposite boundary chain, as required by the definition of street polygons (see e.g. [13] for a more detailed treatment). We call this arrangement left-right consecutive pockets. Notice that in general this property only holds for the points u and v in P, and is not necessarily the case for other points w on the boundary of P.

Definition 5. *A chord between two points $\overline{w_1 w_2}$ on the boundary of the polygon is said to be w_1-minimal if and only if there exists an $\epsilon > 0$ such that for all chords with end points (w_1, w_2') and $|w_2 - w_2'| < \epsilon$ we have $|\overline{w_1 w_2}| < |\overline{w_1 w_2'}|$.*

Notice that w_1-minimal chords either form a right angle with the boundary at w_2, or w_2 is a reflex vertex of P.

A chord $\overline{uw}$ is clasified as left, right or middle depending on its position with respect to the surrounding pockets. That is, if a chord is located between two consecutive left (right) pockets is called a left (right, respectively) chord. If the chord is located between a right and a left pocket, in clockwise order, then it is termed a middle chord.

3 Searching for a Target from a Restricted Starting Point

In this section we consider the problem of searching for a target located at an arbitrary point t in the interior of a street polygon, with the robot starting from the point $s = u$ on the boundary.

Lemma 2. *If c is a chord with endpoints (u, w) in a street polygon (P, u, v), then it splits P into two parts P_1 and P_2, and one of P_1 and P_2 is weakly visible from c while the other contains the point v.*

Proof. Clearly v is contained in one of the two parts, assume that it is on the left side, P_1 (the other case is symmetrical). Therefore the entire counterclockwise polygonal chain from u to w is contained in P_2. Moreover, we know that the polygonal chain from u to w sees the left chain L in P_1. 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 in P_2.

Observation 1 *The point v lies to the right of all but the last left pocket edge and to the left of all but the last right pocket edge.*

Theorem 1. *There exists a strategy for searching for a target of arbitrary location t inside a street (P, u, v) starting from $s = u$ with a competitive ratio of at most 36.806.*

Proof. The proof of this theorem is based on the algorithm for star polygons first presented in [15] and further improved in [19]. However, there are several key differences which result in a significantly larger competitive ratio than the case of a star polygon.

This algorithm traverses left and right pockets edges alternatively, and in order of increasing length, until the entire polygon is seen. The aim of the algorithm is to traverse those pocket edges that form chords, since such chords fully explore a portion of the polygon as shown in Lemma 2. Otherwise, if there is no pocket edge at a given distance forming a chord, then the robot traverses an unexplored pocket edge to its current entrance point. We classify extended pocket edges in two groups, $\mathcal{F}_{left}$ and $\mathcal{F}_{right}$. The robot starts with the subset of currently visible extended pocket edges $\mathcal{F}^0_{left}$ and $\mathcal{F}^0_{right}$, respectively. These sets are updated as the robot explores pocket edges and at the same time discovers new ones. 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.

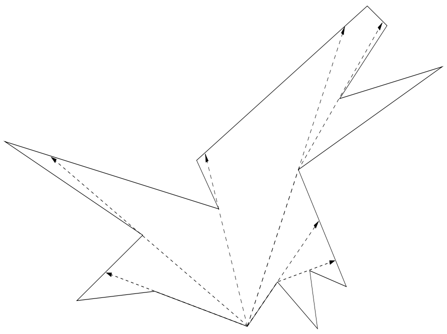

Fig. 3. Extended pocket edge search.

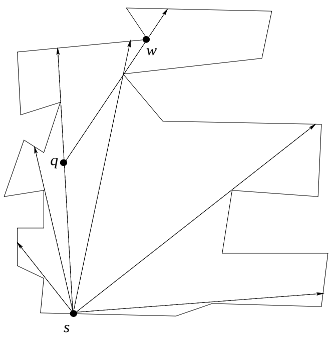

Fig. 4. A left pocket inside a right pocket.

Let *side* $\in$ {*left*, *right*} and if *side* $= right$, then $\neg$*side* $=$ *left* and vice versa, and let $a > 1$ be a constant.

Algorithm Restricted-Start Free-Target Search
Input: A street polygon (P, u, v) and a starting point $s = u$ (notice that the location of v is not required by the algorithm);
Output: The location of the target point t;

let $\mathcal{F}_{left}$ ($\mathcal{F}_{right}$) be the set of extended left (right) pocket edges currently seen but not explored;
(∗ Initially $\mathcal{F}_{left}$ and $\mathcal{F}_{right}$ contain only simple pocket edges; ∗)
let p_E be the closest entrance point to s and E the pocket edge corresponding to p_E;
let d be the distance of p_E to s;
if E is a left pocket edge
  then let *side* $\leftarrow$ *left*
  else let *side* $\leftarrow$ *right*
while $\mathcal{F}_{left} \cup \mathcal{F}_{right}$ is non-empty **do**
  traverse d units on E measuring from s;
  if t is seen **then** exit loop;

```
10        add the new pocket edges seen on this path to F_left or F_right as
              extended pocket edges starting from s;
11        if a new side pocket edge E_N is seen inside a ¬side pocket and
            |E_N| ≤ d
12          then let E ← E_N;
13                move to the entrance point of the ¬side pocket and explore
                  E;
14        remove from F_side all extended side pocket edges to the side side
              of the extended pocket edge E
15        if l_E is reached then remove E;
16        move back to s;
17        let d ← a · d;
18        let side ← ¬side;
19        if side = left
20          then let E ∈ F_left such that p_E is the rightmost entrance point
                  or pocket forming a chord with d(s, p_E) ≤ d.
21                if there is no such edge
22                  then select as E as the leftmost edge in F_left
23          else  (* side = right *)
                  select E analogously;
      end while;
24    move to t;
```

An important difference with the star searching algorithm of [15, 19], is that in this case it is possible for a left pocket to be contained inside a right pocket and vice versa. Figure 4 illustrates one such case, where traversal of a left pocket edge leads to the discovery of a further left pocket edge hidden inside a right pocket.

Assume that the new pocket edge is left and is contained in a right pocket (the other case is symmetric). When the algorithm sees the new pocket edge it adds it to $\mathcal{F}_{left}$. Furthermore, if the length of the new pocket edge is smaller than the one currently being explored, then the robot moves on the new pocket edge. This causes a detour in the algorithm, since if the robot had known of the existence of such hidden pocket, it would have travelled straight to the entrance of the right pocket edge and from there to the hidden left pocket edge. Unfortunately that edge was not in contention in Step 20. The length of this detour can be bounded as follows.

Claim. Let q be the point on the original pocket edge where the robot discovered the new pocket edge, and let w be the entrance to the pocket defined by this new pocket edge. Then $d(s,q) + d(q,w) \leq (2a+1)\, d(s,w)$.

Proof. Since the new left pocket was hidden inside an unexplored right pocket edge, we know that the distance $d(s,w)$ must be larger than the value d used to explore in the last step, as otherwise that pocket would have been explored. Therefore we have that $d(s,q) \leq a\, d \leq a\, d(s,w)$. Now, the robot must reach w from q (see Figure 4). We apply the triangle inequality and obtain $d(q,w) \leq$

$d(s,q) + d(s,w) \leq a\, d(s,w) + d(s,w)$. Therefore the total distance traversed by the robot to reach w is at most $d(s,q) + d(q,w) \leq (2a+1)\, d(s,w)$.

The last observation we need to make is that such a hidden pocket edge discovery might happen more than once within one exploration step. That is, once the robot starts moving towards the newly discovered pocket edge it might discover yet another left pocket edge with entrance w' further inside the right pocket. This reflects the case of a street with more than one "funnel structure" (see for example [13]). Klein showed that since the shortest path to the hidden pocket goes through the entrance of the right visible pocket, the street polygon can be decomposed into a sequence of funnel structures. The search strategy then has a competitive ratio no greater than the maximum of the competitive ratios in each of the consecutive funnel structures [13, 16].

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

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

Clearly the algorithm always terminates, as it either finds the target or it eventually explores all pocket edges. In the later case we must ensure that the target is also found. This follows from Lemma 1, Lemma 2 and Observation 1. Indeed, after exploring the last pocket edge all of the polygon to the left of the last left pocket edge has been explored, all of the polygon to the right of the last right pocket edge has been explored as well and there are no unseen areas (i.e. pockets) left to explore. Therefore, the target must have been discovered in the last step when the robot reaches the entrance point of the last pocket edge and in all cases the target is found.

The competitive ratio is derived from the Claim 3 and the Invariant. After Step 16, the invariant holds because if there was a, say, left pocket at a distance of less than d/a^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/a^2 which is to the right of it was traversed. But exploring this second edge entails exploring the earlier edge as shown in Lemma 2 and Observation 1.

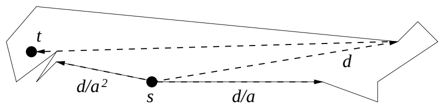

Fig. 5. Worst case discovery of a target.

This means that after Step 16 we know that the target must be located inside a pocket with entrance point at a distance of strictly greater than d/a^2 from s. The worst case occurs when the robot sees the target at a distance of $d/a^2 + \epsilon$, at the very end of a search of length d (see Figure 5). This means that the ratio of the distance traversed by the robot according to Algorithm *Restricted-Start Free-Target Search* to the distance from s to t is at most

$$2\frac{\sum_{i=0}^{n}(2+1/a)\,a^{i}}{a^{n-2}}+1=2\frac{a^{2}\,(2a+1)}{a-1}+1.$$

This expression is minimized when $a=(5+\sqrt{57})/8$ which gives a competitive ratio of $(151+19\sqrt{57})/8 \leq 36.806$ as claimed.

4 Searching in a Street from an Arbitrary Starting Point

In this section we present an algorithm for searching for a target of arbitrary position t in a street polygon, starting from an arbitrary point s on the boundary of the polygon. In other words we remove the restriction from the previous section that $s=u$. Moreover, the robot does not need to know the location of u and v to explore P. The algorithm is considerably more involved than the one for restricted starting position, and the competitive ratio is somewhat larger, as it is to be expected.

A chord c inside a street polygon (P,u,v) splits a polygon in two parts. P_1, and P_2. The points u or v may be located both on one of the two parts, or one on each part.

Lemma 3. *Consider a chord c in P and assume that u and v are on the same side of P, say P_1. Then P_2 is weakly visible from c.*

Proof. Since both u and v are in P_1, one of the two polygonal chains from u to w is entirely contained in P_1, say the left polygonal chain L from u to v. We know that any point on $R\cap P_2$, where R is the right polygonal chain from u to v, sees at least one point in L. But any line contained in the polygon and joining a point in P_1 with a point in $R\cap P_2$ intersects the chord c. This implies that the chord weakly sees all points in P_2.

Notice that this lemma holds for any simple path between two points on the boundary of the polygon (not just a chord) as long as the points u and v are on the same side of the path.

Definition 6. *If the points u and v are one on each side of the chord c, say u in P_1 and v in P_2, then the chord splits each of L and R in two. Let $L_L=L\cap P_1$, $L_R=L\cap P_2$, $R_L=R\cap P_1$ and $R_R=R\cap P_2$.*

For example, in Figure 6, the chord $c=(s,w)$ splits the left (clockwise) polygonal chain L from u to v in two parts, from u to w and from w to v corresponding to L_L and L_R, respectively. Similarly, the counterclockwise chain R from u to v is split into two parts from u to s and from s to v which correspond to R_L and R_R, respectively.

Lemma 4. *In the polygon formed by the chord $c=(s,w)$, L_L and R_L, there are only right pockets on L_L and only right pockets in R_L visible from c. Similarly, for the polygon formed by c, L_R and R_R, there are only right pockets on R_R and left pockets on L_R visible from c.*

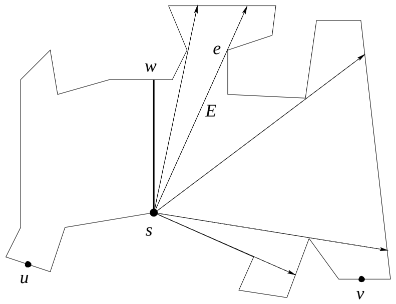

Fig. 6. Right pockets in L_R.

Proof. Assume otherwise. That is, there is, say a right pocket on L_R with an associated pocket edge E with entrance point e (see Figure 6). In this case, the boundary points delimiting the window edge of this pocket are both in L. Since the polygon is a street, every point in the boundary of the pocket can see at least one point in R. Now, the two edges (w, s) and (s, e) together form a path separating the points in the pocket from the right polygonal chain R. Therefore, from Lemma 3 it follows that the pocket must be entirely visible from this path. Since the pocket cannot be seen from the pocket edge itself, the pocket must be visible from the chord c, which is a contradiction. The same argument applies to L_L. For R_R and R_L, the robot moves to w and the argument above also applies.

A general description of the algorithm is to traverse edges as in the Restricted-Start Free-Target Search (RSFTS) algorithm described in the previous section as long as the pocket edges are left-right consecutive and the entire portion of the polygon to the left side of a left pocket edge (and to the right side of a right pocket edge) can be seen from the edge. If the portion of the polygon to the left of a left pocket edge already explored was not seen in its entirety, we know by Lemma 3 that u and v must necessarily be on opposite sides of this pocket edge and Lemma 4 applies. The same holds for the right part of the polygon to the right of a pocket edge. If the pocket edges are no longer left-right consecutive, the robot selects the shortest length minimal middle chord and traverses it, which splits the polygon in two parts. In this case, since the chord is of type middle, it follows that the points u and v must be on opposite side of the chords, and therefore we are in the situation described in Lemma 4 and Definition 6.

In either case, we are in the situation of Lemma 4 and the robot simply searches each side using the RSFTS algorithm. The competitive ratio corresponds then to a four ray search, which gives a different choice of a for RSFTS. More formally,

Theorem 2. *There exists a* 69.216*-competitive strategy that finds a target of arbitrary position in a street polygon starting from a point s on the boundary.*

Proof. The algorithm is a modification of RSFTS. Initially the robot executes lines 1-21 of RSFTS with the exception of line 7 which now reads:

7 **while** $\mathcal{F}_{left} \cup \mathcal{F}_{right}$ is non-empty **and** the pocket edges appear in consecutive left-right order **and** all the *side* of a *side* pocket edge was seen **do**

Lines 22 onwards are replaced by

```
22 if t was found then move to t;
   (* Since we exited the loop without finding t, pockets are not in left-right order *)
23 let M be the set of minimal middle chords;
24 sort M by increasing length;
25 traverse the chord c ← min(M);
26 the chord c splits P in two parts. Let P1 and P2 be those parts;
27 while target has not been found do
28     alternatingly apply one step of RSFTS on P1 and P2;
29 endwhile
30 move to t;
```

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/a^4 or less.

The correctness of the algorithm follows from Lemmas 2-4 and Observation 1. Lemmas 3 and 4 guarantee that either we can explore the entire polygon using RSFTS or the polygon is split into two pieces. Lemma 2 and Observation 1 guarantee that each of the parts can be explored using RSFTS.

As before the worst case competitive ratio occurs when the target is located at a distance $d/a^4 + \epsilon$, and the competitive ratio is given by

$$2\frac{\sum_{i=0}^{n}(2+1/a)\,a^i}{a^{n-4}} + 1 = 2\frac{a^4\,(2a+1)}{a-1} + 1.$$

This expression is minimized when $a = (7+\sqrt{177})/16$ which gives a competitive ratio of $(71893 + 5251\sqrt{177})/2048 \leq 69.216$ as claimed.

5 Lower Bounds

In Figure 7 we have a street polygon that provides a 9 lower bound on the competitive ratio of searching in streets starting from a point $s = u$. This polygon can be explored, say, by traversing the path (u, v) from which, by definition, the entire polygon is seen. Notice that from each indentation we can see the opposite polygonal chain somewhere in the upper part of the polygon. As we increase the height of the polygon and make the angle of the walls of each indentation go to $\pi/2$ the polygon remains a street, yet traversing (u, v) is no longer an efficient exploration strategy. Thus the robot is restricted to exploring the base using a doubling strategy, which has a 9 competitive ratio (see [2, 1, 7] for a lower bound on doubling and [15, 19] for a more detailed analysis on this general type of indented rectangular polygons).

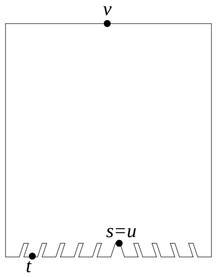

Fig. 7. Lower bound when $s = u$.

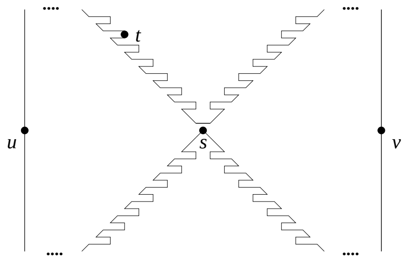

Fig. 8. Lower bound for arbitrary position.

For the case of searching from an arbitrary position the polygon of Figure 8 is a street. In this case the indentations along the diagonals, which seem to be horizontal, are in fact slanted just enough to actually intersect the vertical edges on the opposite side of the street. For example, the extension of the nearly horizontal walls of the indentation containing t in the figure above intersect the left vertical line just below u. As the distance from u to v is increased, the angle of the walls of the indentations goes to zero. In this case the robot is forced to do a simplified form of a four ray search, which can be shown to have competitive ratio of at least $a^4/(a-1)+a^3$. This is minimized for $a = (5+\sqrt{7})/6$ with competitive ratio of at least 11.78.

6 Conclusions

We have presented a strategy for on-line searching of a street polygon regardless of the starting position of the robot or the location of the target. The strategy proposed has a constant competitive ratio. This is in contrast to previous strategies for searching on streets as well as other classes of polygons for which the choice of position of the target and the starting position are highly restricted in order to achieve a constant competitive ratio. We provided lower bounds for this problem.

We also presented a more efficient strategy for the special case when the robot starts from a distinguished point on the polygon but the target is free to select its hiding position, and gave a lower bound for this variant as well.

Acknowledgements: We wish to thank Sven Schuierer for helpful discussions on this subject.

References

1. R. Baeza-Yates, J. Culberson and G. Rawlins. "Searching in the plane", *Information and Computation*, Vol. **106**, (1993), pp. 234-252.
2. A. Beck. "On the linear search problem", *Israel Journal of Mathematics*, Vol. **2**, (1964), pp. 221-228.

3. B. Bhattacharya and S. K. Ghosh, "Characterizing LR-visibility polygons and related problems", *Proc. 10th Canadian Conference on Comp. Geom.*, (1998).
4. C. A. Bröcker and S. Schuierer, "Searching Rectilinear Streets Completely", In these *Proceedings.*
5. A. Datta, Ch. Hipke, and S. Schuierer. "Competitive searching in polygons—beyond generalized streets", in *Proc. 6th Int. Symposium on Algorithms and Computation (ISAAC)*, 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 (SoCG)*, (1994), pp. 175-182.
7. S. Gal. *Search Games*, Academic Press, 1980.
8. Ch. Icking. "Motion and visibility in simple polygons". Ph.D. Thesis, Fernuniversität Hagen, 1994.
9. Ch. Icking and R. Klein. "Searching for the kernel of a polygon. A competitive strategy", *Proc. 11th ACM Symposium on Computational Geometry (SoCG)*, (1995).
10. Ch. Icking, R. Klein and E. Langetepe. "An Optimal Competitive Strategy for Walking in Streets." *Tech.Rep. 233*, Dept. of Comp. Sci., FernUniversität Hagen, Germany, 1998.
11. B. Kalyasundaram and K. Pruhs. "A competitive analysis of algorithms for searching unknown scenes", *Comp. Geom.: Theory and Applications*, Vol. **3**, (1993), pp. 139-155.
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 (SODA)*, (1993), pp. 441-447.
13. R. Klein. "Walking an unknown street with bounded detour", *Comp. Geom.: Theory and Applications*, Vol. **1**, (1992), pp. 325-351, also in *32nd FOCS*, 1991.
14. J. Kleinberg. "On-line search in a simple polygon", *Proceedings of 5th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, (1994), pp. 8-15.
15. A. López-Ortiz. "On-line target searching in bounded and unbounded domains", Ph.D. thesis, University of Waterloo, 1996.
16. A. López-Ortiz and S. Schuierer. "Going home through an unknown street", *Proc. 4th Workshop on Data Structures and Algorithms (WADS)*, 1995, Lecture Notes in Computer Science 955, Springer-Verlag, pp. 135-146.
17. A. López-Ortiz and S. Schuierer. "Generalized streets revisited", In J. Diaz and M. Serna, editors, *Proc. 4th European Symposium on Algorithms (ESA)*, LNCS 1136, pages 546–558. Springer Verlag, 1996.
18. A. López-Ortiz and S. Schuierer. "Walking streets faster", *Proc. 5th Scandinavian Workshop in Algorithmic Theory Algorithms (SWAT)*, 1996, Lecture Notes in Computer Science 1097, Springer-Verlag, pp. 345-356.
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. S. Schuierer and I. Semrau. "An optimal strategy for searching in unknown streets." *Proc. 16th Symp. on Theoretical Aspects of Computer Science*, to appear, 1999.