Optimality and Competitiveness of Exploring Polygons by Mobile Robots

Jurek Czyzowicz*, Arnaud Labourel**, and Andrzej Pelc***

Département d'informatique, Université du Québec en Outaouais,
Gatineau, Québec J8X 3X7, Canada.
jurek@uqo.ca,labourel.arnaud@gmail.com,pelc@uqo.ca

Abstract. A mobile robot, represented by a point moving along a polygonal line in the plane, has to explore an unknown polygon and return to the starting point. The robot has a sensing area which can be a circle or a square centered at the robot. This area shifts while the robot moves inside the polygon, and at each point of its trajectory the robot "sees" (explores) all points for which the segment between the robot and the point is contained in the polygon and in the sensing area. We focus on two tasks: exploring the entire polygon and exploring only its boundary. We consider several scenarios: both shapes of the sensing area and the Manhattan and the Euclidean metrics.

We focus on two quality benchmarks for exploration performance: optimality (the length of the trajectory of the robot is equal to that of the optimal robot *knowing* the polygon) and competitiveness (the length of the trajectory of the robot is at most a constant multiple of that of the optimal robot knowing the polygon). Most of our results concern rectilinear polygons. We show that optimal exploration is possible in only one scenario, that of exploring the boundary by a robot with square sensing area, starting at the boundary and using the Manhattan metric. For this case we give an optimal exploration algorithm, and in all other scenarios we prove impossibility of optimal exploration. For competitiveness the situation is more optimistic: we show a competitive exploration algorithm for rectilinear polygons whenever the sensing area is a square, for both tasks, regardless of the metric and of the starting point. Finally, we show a competitive exploration algorithm for arbitrary convex polygons, for both shapes of the sensing area, regardless of the metric and of the starting point.

1 Introduction

The model and the problem. A mobile robot, represented by a point moving along a polygonal line in the plane, has to explore an unknown polygon and

* Partially supported by NSERC discovery grant.

** This work was done during this author's stay at the Université du Québec en Outaouais as a postdoctoral fellow.

*** Partially supported by NSERC discovery grant and by the Research Chair in Distributed Computing at the Université du Québec en Outaouais.

return to the starting point. We assume that the boundary is included in the polygon. The robot has a *sensing area* (abbreviated by SA in the sequel) which can be a circle or a square centered at the robot. During the exploration the robot must remain within the polygon, but its SA can partially exceed the boundaries of the polygon. At each point of its trajectory the robot "sees" (explores) all points for which the segment between the robot and the point is contained in the polygon to be explored and in the sensing area. For any explored point the robot is aware of whether this point is on the boundary of the polygon or not. We consider two tasks: exploring the entire polygon and exploring its boundary, for both shapes of the SA and for the Manhattan and the Euclidean metrics. The Manhattan metric will be called L_1 and the Euclidean metric will be called L_2 (Recall that in the L_1-metric the distance between two points is the sum of the differences of their coordinates). We also differentiate the situation when the starting point of the robot is at the boundary and when it is an arbitrary point of the polygon. We assume that the robot remembers what it has explored, i.e., it keeps a partial map of the explored part of the polygon with its trajectory in it, at all times.

The quality measure of an exploration algorithm not knowing the polygon (an on-line algorithm) is the length of the trajectory of the robot, and we seek to minimize this length. We compare it to the smallest length of the trajectory of a robot *knowing* the polygon (an off-line algorithm), executing the same task (exploring the boundary or exploring the entire polygon) and starting at the same point. The ratio between these two lengths, maximized over all pairs (polygon, starting point), is the *competitive ratio* of the on-line exploration algorithm. We focus on two quality benchmarks for exploration performance: optimality (competitive ratio equal 1) and competitiveness (constant competitive ratio).

Our results. Our first set of results concerns the possibility of optimal on-line exploration. Here we consider only rectilinear polygons (those whose angles are either $\pi/2$ or $3\pi/2$). It turns out that optimal exploration is possible only in one scenario, that of exploring the boundary by a robot with square sensing area aligned with the sides of the polygon, starting at the boundary and using the L_1-metric. For this case we give an optimal exploration algorithm. In all other scenarios (when *either* the entire polygon has to be explored, *or* the sensing area is a circle, *or* the metric is L_2, *or* the starting point may be strictly inside the polygon) we prove impossibility of optimal on-line exploration.

For competitiveness, the situation is more optimistic: our optimal boundary exploration algorithm yields a competitive exploration algorithm for rectilinear polygons whenever the sensing area is a square aligned with the sides of the polygon, for both tasks (exploring the boundary or the entire polygon) regardless of the metric and of the starting point. Finally, we show a competitive exploration algorithm for arbitrary convex polygons, for both shapes of the sensing area, regardless of the metric and of the starting point.

To the best of our knowledge we propose the first competitive on-line algorithm to explore arbitrary rectilinear polygons with some limited sensing area.

Related work. Exploration of unknown environments by mobile robots was extensively studied in the literature under many different models. One of the most important works in this domain is [5] where the sensing area is unlimited. The authors gave a 2-competitive algorithm for rectilinear polygon exploration. The competitive ratio was later improved to 5/3 in [8]. It was shown in [13] that there is no deterministic algorithm for this problem better than 5/4-competitive and that there exists a 5/4-competitive randomized algorithm solving it. All these results hold for the L_1-metric. Upper bounds for the L_2-metric can be obtained from the fact that any α-competitive algorithm for the L_1-metric is $\alpha\sqrt{2}$-competitive for the L_2-metric [5]. The case of non-rectilinear polygons was also studied in [4, 10] and a competitive algorithm was given in this case.

For polygonal environments with an arbitrary number of polygonal obstacles, it was shown in [5] that no competitive strategy exists, even if all obstacles are parallelograms. Later, this result was improved in [1] by giving a lower bound in $\Omega(\sqrt{k})$ for the competitive ratio of any on-line algorithm exploring a polygon with k obstacles. This bound remains true even for rectangular obstacles. Nevertheless, if the number of obstacles is bounded by a constant m, then there exists a competitive algorithm with competitive ratio in $O(m)$ [4].

Exploration by a robot with a limited sensing area has been studied, e.g., in [6, 7, 11, 12, 15]. This model is interesting to study, since it is justified by real world constraints. Indeed, computer vision algorithms based on information obtained by sensors, such as stereo or structured-light finder, can reliably compute visibility scenes only up to a limited range [7]. To the best of our knowledge, there were no previous results concerning competitive on-line exploration for *arbitrary* rectilinear polygons with limited visibility.

The off-line exploration problem with limited SA is related to older problems such as *lawn mowing*, *pocket milling* and *ice rink* problems. All these three problems are concerned with finding an optimal path of a tool moving on a surface (grass area to mow, pocket to mill or ice rink to sweep), such that all points of the surface are covered by the tool (a mower, cutter or ice rink machine) at least once during its travel. The only difference between exploration and the lawn mowing problem is that the robot is not allowed to leave the environment, while the mower can exit the surface. The ice rink problem is the same as the lawn mowing problem, except for the notion of the optimal path. In lawn mowing, only the length of the path is considered, while in the ice rink problem we also need to take into account the number of turns done by the robot, since those turns are costly [14]. In the pocket milling problem, not only the robot cannot leave the surface but also the cutter must not leave it. Here, the goal is to find a shortest path that covers the maximum area possible. The first two problems are NP-hard and the complexity of the third one is unknown [9]. All three problems admit polynomial time approximation algorithms [2, 14].

On-line exploration with limited SA has been studied, e.g., in [6, 11, 12]. Unlike in our model, the robot in [6] can see slightly farther than its tool (six times the tool range). The authors describe an on-line algorithm with competitive ratio $1+3(\Pi D/A)$, where Π is a quantity depending on the perimeter of P, D the size of the tool and A the area of P. Since the ratio $\Pi D/A$ can be arbitrarily large, their algorithm is *not* competitive in the general case. Moreover, the exploration in [6] fails on a certain type of polygons, such as those with narrow corridors.

In [11, 12], the authors consider the exploration of a particular class of polygons: those composed of complete identical squares, called cells of size a priori known to the robot. In this model, the robot explores all points in a cell when it enters the cell for the first time, and can move in one step to any adjacent cell. The cost of the exploration is measured by the number of steps. There exists a 2-competitive algorithm for exploration of such polygons with obstacles [11]. For polygons without obstacles, there exists a 4/3-competitive algorithm for exploration and no algorithm can achieve a competitive ratio better than 7/6 [12].

There are only a few papers on how to explore the boundary of a terrain with limited sensing area. This problem was first considered in [15] (in its off-line version) using a reduction to the safari route problem. The *safari route problem* consists in finding a shortest trajectory, starting at the point s of the boundary of a polygon P and going back to s, that visits a specified set of polygons $\mathcal{P}$ contained in P. It is assumed in [15] that the polygons in $\mathcal{P}$ are attached to the boundary of P (share at least one point with the boundary of P), since otherwise the problem is NP-hard [15]. The author gives a $O(mn^2)$ algorithm solving this problem, where m is the cardinality of $\mathcal{P}$ and n is the total number of vertices of P and polygons in $\mathcal{P}$. It is shown that an optimal safari route visiting all the circular sectors of vertices corresponding to the angles of P, (i.e., the region inside P from which the vertex is visible), is an optimal boundary exploration trajectory [15]. To solve the safari route problem, circular sectors are approximated with polygons and the obtained solution is within 0.3% of optimal. It is computed in cubic time.

2 Definitions and preliminary results

In this section and in the part of the paper concerning optimality of exploration, we only consider rectilinear polygons. Let P be such a polygon. For convenience, without loss of generality, we assume that all sides of the polygon P are either parallel to the x-axis (east-west sides) or to the y-axis (north-south sides).

A *rectilinear* trajectory path has all its segments parallel to either the x-axis or the y-axis. Since in the L_1-metric there is always a rectilinear path among the shortest paths between two points, we consider only rectilinear paths and we drop the word "rectilinear" in all considerations regarding the L_1-metric. In particular, we use this convention in this section and in Section 3.1.

A segment T contained in a polygon P is *separating*, if it divides P into two simple polygons called the *subpolygons defined by* T. The *foreign polygon* defined by T according to a point u, denoted by $FP_u(T)$, is the subpolygon not containing u. Note that the foreign polygon is undefined if $u \in T$. A separating segment T *dominates* a separating segment T' according to the point u, if $FP_u(T)$ is strictly contained in $FP_u(T')$.

The robot at position r *explores* a point x, if the segment $\overline{rx}$ is included both in the polygon and in the SA centered in r. We consider two types of SA: a *round* SA which is a disc of diameter 2 and a *square* SA which is a 2×2 square. For exploration of rectilinear polygons, we assume that the sides of a square SA are aligned with the sides of the polygons. An exploration trajectory of polygon P is a path contained in P such that each point of P is explored by the robot at some point of this path. A *boundary exploration trajectory* is a trajectory of a robot inside the polygon P, exploring the boundary of P. In both cases, the start and the end of the trajectory are equal and are denoted by r_0.

For each side S of a polygon P, we extend S inside P, possibly from both ends, until it first hits the boundary of P. Each contiguous section of the resulting segment, if any, excluding S itself, is called an *extension segment* (cf. [5]) associated with S. For each side S of a polygon P, we draw the line L parallel to S at distance one from it, on the side of the interior of P. If this line intersects P, we define the *vicinity segment* associated with S, as the part of L between the closest point of $P \cap L$ from S in clockwise order along the boundary and the closest one in anti-clockwise order.

Lemma 1. *Any boundary exploration trajectory is not shorter than GE.*

Each extension or vicinity segment M of side S is a separating segment. In the rest of the paper, any domination relation or foreign polygon $FP(M)$ is defined according to point r_0, if no other point of reference is specified. If $r_0 \in M$, we set $FP(M)$ to be the subpolygon defined by M that contains S. Starting at r_0, if side $S \in FP(M)$, where M is an extension or vicinity segment of S, then S can become explored only if M is visited (i.e., either crossed or touched). If this is the case, we call M a *necessary segment* of S. For two necessary (extension or vicinity) segments M_1 and M_2, if M_1 dominates M_2 then there is no way to visit M_1 without crossing M_2 from r_0. So, we can ignore M_2, since it is automatically visited, if we visit M_1. A non-dominated necessary segment is called *essential*. To see all sides of a polygon, starting at r_0, the robot has to visit every essential segment.

If the starting point r_0 is on the boundary of P, then it induces a *natural order* of essential segments, clockwise along the boundary of the polygon P: $E_1, E_2, \ldots, E_m$, where E_1 is the first essential segment encountered when moving clockwise along the boundary from r_0, and so on. For $i \in 1, \ldots, m$, we denote by x_i the point on E_i at the minimum distance from point x_{i-1}, with the starting point $r_0 = x_0$. As shown in [5], these points are uniquely defined by r_0. This trajectory from x_0 to x_m, and back directly from x_m to x_0, is called GE for 'Greedy Essential'.

3 Optimality

3.1 The optimal boundary exploration algorithm

In this section, we assume that the SA is a 2×2 square aligned with sides of a rectilinear polygon. Our aim is to construct a boundary exploration algorithm starting at a boundary point r_0 and following GE as closely as possible. Unfortunately, in the case of a robot with bounded SA (unlike the robot from [5] which had unbounded visibility) it is impossible for an on-line algorithm to visit essential extensions greedily, using shortest paths. The following proposition shows this significant difference between our scenario and that from [5].

Proposition 1. *There is no on-line algorithm that greedily visits the essential segments of every polygon, i.e., that visits the essential segments by following shortest paths between them, even starting at the boundary.*

Since, as shown above, our bounded visibility scenario is more difficult than that from [5], our optimal boundary exploration algorithm must also be more subtle. Its idea is as follows.

The robot tries to increase the contiguous part of the boundary seen to date. The rest of the boundary is not yet explored by the robot for three possible reasons: an obstructing angle limiting the view of the currently explored side, a $3\pi/2$-angle terminating the currently explored side and obstructing the view of the next side, or finally the end of the SA limiting the view of the currently explored side. The strategy of the robot is to move towards the extension corresponding to the obstructing angle (in the first two cases) and to move parallel to the currently explored side (in the third case). Due to limited visibility, no necessary segment is seen by the robot in the third case, which is a crucial difference between our scenario and that from [5]. While it is impossible to move between consecutive essential segments using shortest paths, we prove that for every essential segment there is some essential segment following it (not necessarily the next one) which the robot reaches by a shortest of all paths visiting the intermediate essential segments. Proving this property is the crucial and technically most difficult part of the algorithm analysis.

Algorithm `BOUNDARY-ON-LINE-EXPLORATION` (*BOE*, for short)
INPUT: A starting point r_0 on the boundary of the polygon to be explored.
OUTPUT: A shortest boundary exploration trajectory, starting and ending at r_0.

We denote by C the contiguous part of the boundary, starting clockwise from r_0, that has been explored so far by the robot, and we call *frontier*, denoted by f, the end of C. The current position of the robot is denoted by r.

Repeat the following strategy until C becomes the boundary of a simple polygon, updating r, f and C whenever any change occurs.

Case 1: *There is an obstructing angle b, i.e., r, b and f are aligned and b is a $3\pi/2$ angle not in C (see Fig. 1(a))*
Move towards the extension $E(b)$ of the side $U(b)$ incident to b and not explored from r. The strategy used to reach $E(b)$ is to move parallel to the other side $S(b)$ incident to b whenever possible, and move towards $S(b)$, parallel to $E(b)$, until it becomes possible again to move parallel to $S(b)$, otherwise.

Case 2: *f is a $3\pi/2$ angle and r is not on the extension $E(f)$ of the side $U(f)$ incident to f and not explored from r. (see Fig. 1(b))*
Same as Case 1 with f instead of b.

Case 3: *There is no obstructing angle, and either f is a $3\pi/2$ angle and r is on the extension $E(f)$ of the side $U(f)$ incident to f and not explored from r, or f is not a $3\pi/2$ angle. (see Fig. 1(c))*

If f is a $3\pi/2$ angle then $S(f) = U(f)$, otherwise $S(f)$ is the side containing f. Move parallel to the side $S(f)$ towards f until:

Case 3.1: *Condition of Case 1 occurs*
Follow Case 1.
Case 3.2: *Condition of Case 2 occurs*
Follow Case 2.
Case 3.3: *A new $\pi/2$ angle a is explored and belongs to C (the robot reaches the vicinity segment $V(a)$ of the new side $U(a)$ incident to a)*
Do nothing (the algorithm proceeds to the next iteration of the repeat loop).

When the above **Repeat** loop is completed (C is a simple polygon), follow a shortest path to r_0 and stop.

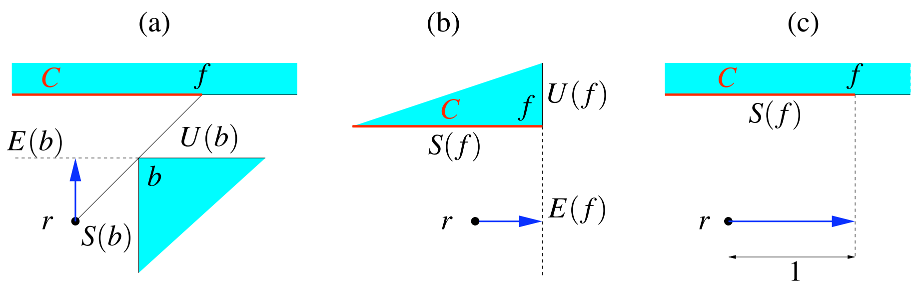

Fig. 1. The three possible configurations during the execution of Algorithm *BOE*

The main result of this section is that Algorithm *BOE* is optimal.

Theorem 1. *Algorithm* `BOUNDARY-ON-LINE-EXPLORATION` *is an optimal on-line algorithm for the boundary exploration of rectilinear polygons with square SA in the L_1-metric, starting and ending at a point of the boundary.*

First, we show that Algorithm *BOE* eventually terminates.

Lemma 2. *Algorithm BOE eventually terminates with C set to the boundary of the input polygon P.*

Let l be the number of iterations of the main loop of Algorithm *BOE* before terminating. For $i = 1, 2, \ldots, l$, the robot is at point r_i at the end of the i-th iteration of the main loop. The point r_i is either on a vicinity or on an extension segment denoted by M_i. Indeed, at the end of an iteration corresponding to Cases 1 or 3.1, the robot is on the extension segment $E(b)$ of side $U(b)$. For Cases 2 or 3.2, the robot is on the extension segment $E(f)$ of side $U(f)$. Finally, for Case 3.3, the robot is on the vicinity segment $V(a)$ of side $U(a)$.

We define a new trajectory BOE' that reaches segments M_i in a greedy way. For $i \in 1, \ldots, l$, we denote by z_i the point on M_i at the minimum distance (in the L_1-metric) from point z_{i-1}, with $r_0 = z_0 = z_{l+1}$. More formally, the trajectory BOE' is the one following a shortest path from z_{i-1} to z_i, for all $1 \leq i \leq l + 1$.

Although BOE might not follow a shortest path between the segment M_i and M_{i+1} for some i, its total length turns out to be equal to that of BOE'. We denote by $BOE[r_i, r_k]$ (resp. $BOE'[z_i, z_k]$) the part of the trajectory BOE (resp. BOE') between the points r_i and r_k (resp. z_i and z_k).

Lemma 3. *The BOE' trajectory has the same length as the BOE trajectory.*

Proof. We show that for all i, there exists a j, such that $BOE[r_i, r_{i+j}]$ is a shortest path from point r_i to M_{i+j} that visits segments M_{i+k} for $1 \leq k < j$. The proof depends on the type of the $(i+1)$-th iteration of Algorithm BOE.
Case 1: The robot follows a shortest path from r_i to the extension segment $E(b) = M_{i+1}$ as shown in [5]. Hence, the property holds for $j = 1$.
Case 2: Same as Case 1 with f instead of b.
Case 3.1: The robot moves parallel to $S(f)$ and then moves towards the extension segment $E(b)$, where b obstructs the vision to $S(f) = \overline{dv}$ (with d the first vertex of $S(f)$ in clockwise order) from the robot. Assume, without loss of generality, that the robot moves east when moving parallel to $S(f)$ ($S(f)$ is an east-west side) and is south of $S(f)$.

In order to explore the vertex v, the robot has to execute an iteration corresponding to Cases 2, 3.2 or 3.3. Let j denote the number of iterations executed by the robot to fully explore $S(f)$, the last one corresponding to Case 2, 3.2 or 3.3, needed to explore v.

Let S' be the side following the side $S(f)$ in the clockwise order. The segment M_{i+j} is either the extension segment of S', if the angle between $S(f)$ and S' is a $3\pi/2$ angle, or the vicinity segment of S', if the angle between $S(f)$ and S' is a $\pi/2$ angle. In both cases, M_{i+j} is perpendicular to all M_{i+k} for $0 \leq k < j$ and is east of point r_{i+1}.

During the iterations corresponding to Cases 1 or 3.1, the robot moves either north or east, since for all $1 \leq k < j$, M_{i+k} is an east-west segment and the obstructing angle b_k is in the north-east quadrant of the SA of the robot. During the last iteration corresponding to Cases 2, 3.2, or 3.3, the robot moves either north or east, since M_{i+j} is a north-south segment and the angle f (Cases 2 or 3.2) or the angle a (Case 3.3) is in the north-east quadrant of the SA of the robot. Hence, the path from r_i to r_{i+j} is a shortest path.

We show that the point r_{i+j} is the point of M_{i+j} at minimal distance from r_i. Indeed, it is reached by minimal x-axis and y-axis shifting, since the path is monotone and the robot moves north only until reaching the y-coordinate of the angle b_{j-1}. Hence, the path is a shortest path to M_{i+j} visiting all M_{i+k} for $1 \leq k < j$, and the property is verified.
Case 3.2: The robot moves parallel to $S(f)$ and then applies the strategy of Case 2. Since this strategy consists in moving parallel to $S(f)$ whenever it is possible, the property is verified, as in Case 2.
Case 3.3: The robot moves parallel to $S(f)$ from r_i to the vicinity segment M_{i+1} of the side immediately after $S(f)$ in clockwise order. The path followed by the robot to reach M_{i+1} is a shortest path, since $S(f)$ is perpendicular to M_{i+1}. Hence, the property holds for $j = 1$.

Recall that $r_0 = z_0$. We showed that $r_i = z_i$ implies $|BOE[r_i, r_{i+j}]| = |BOE'[z_i, z_{i+j}]|$, and $r_{i+j} = z_{i+j}$, for the index j (depending on i) determined above, since $BOE[r_i, r_{i+j}]$ is a shortest path from r_i to M_{i+j}. It follows by induction that $|BOE| = |BOE'|$.

We define a *compatible order* of essential segments as follows. In the natural order of essential segments we choose an arbitrary set of disjoint pairs of consecutive intersecting essential segments, and we swap segments in each pair.

Lemma 4. *The essential segments are visited in a compatible order $D_1, \ldots, D_m$ by the BOE' trajectory.*

In order to compare the BOE trajectory to the GE trajectory, we define a trajectory GC that greedily visits essential segments in the same compatible order as BOE. For $i \in 1, \ldots, l$, we denote by y_i the point on D_i at the minimum distance from point y_{i-1}, with $r_0 = y_0 = y_{m+1}$. More formally, the trajectory GC is the one following a shortest path from y_{i-1} to y_i, for all $1 \leq i \leq m+1$.

Lemma 5. *a) The GC trajectory has the same length as GE.*
b) The BOE' trajectory has the same length as the GC trajectory.

Proof of Theorem 1. Any boundary exploration trajectory (including the optimal one) has length not smaller than that of GE, by Lemma 1. By Lemma 5 (a), we have $|GE| = |GC|$. By Lemma 5(b), we have $|BOE'| = |GC|$. By Lemma 3, we have $|BOE'| = |BOE|$. By Lemma 2, BOE is a boundary exploration trajectory. Hence BOE is an optimal boundary exploration trajectory. □

3.2 Negative results

In this section we show that in all scenarios except the one covered by Theorem 1, optimal on-line exploration is impossible.

Lemma 6. *There is no optimal on-line algorithm for the exploration of rectilinear polygons, with a square SA, in the L_1-metric, even with the starting point at the boundary.*

Proof. We consider two polygons W and T depicted in Fig. 2, and the exploration problem starting from the point x at the boundary of each of these polygons.

Notice that the visible parts of the two polygons are identical when the robot is at any point inside the rectangle $abkl$, the boundary of the rectangle included. So, the adversary can arbitrarily choose one of the two polygons when the robot leaves this rectangle to explore the rest of the polygon. The adversarial strategy to prevent optimality consists in taking the polygon T, if the robot exits the rectangle $abkl$ through point k or b, and in taking the polygon W otherwise.

We first show that any exploration trajectory passing through point b or k in polygon T is not optimal. Note that the order in which the two angles f and g are explored does not matter because of the symmetry of polygon T. The exploration trajectory R of T depicted in Fig. 2 is optimal, since the trajectory follows shortest paths to explore the angle f (at point y) and then the angle g (at point z) starting from point x.

Let us assume, for contradiction, that there is an optimal exploration trajectory E passing through b. In order to have the same length as R, the trajectory E must follow shortest paths from x to b, from b to y, from y to z and from z to x. Let us consider the square region Q of points at distance at least one from lines ab and bk, and at distance at least two from lines lk and gf. The interior points in Q and the points of side kl cannot be explored by a robot following a shortest path xb, by or xy, since these points are at distance larger than one from any shortest path connecting these pairs of points. Consequently, the points of Q and those in the side kl need to be explored when moving on the trajectory between z and x. To explore the points of Q, the robot has to move past the line kl and continue moving east for a distance strictly greater than one. From the fact that this must be a shortest path to x, the robot cannot move west after this move and so cannot explore points of the side kl. Hence, the trajectory E is not an exploration trajectory and so there is no optimal exploration trajectory passing through b. By symmetry of the polygon, the same is true for point k.

We now show that any optimal exploration trajectory passing through any point t of the segment bk (ends of the segment excluded) in polygon W exits the rectangle $abkl$ through b or k. Note that any optimal trajectory needs to explore the angles e or h before the angles f or g. Indeed, there is a shortest path from the point t to a point from which f is visible (respectively the angle g) that explores the angle e (respectively the angle h). Consequently, any optimal exploration trajectory needs to follow an optimal path from t to explore one of the two angles e or h. These paths exit the rectangle $abkl$ through point b or k, and so no optimal exploration trajectory can exit this rectangle through an inside point of the segment.

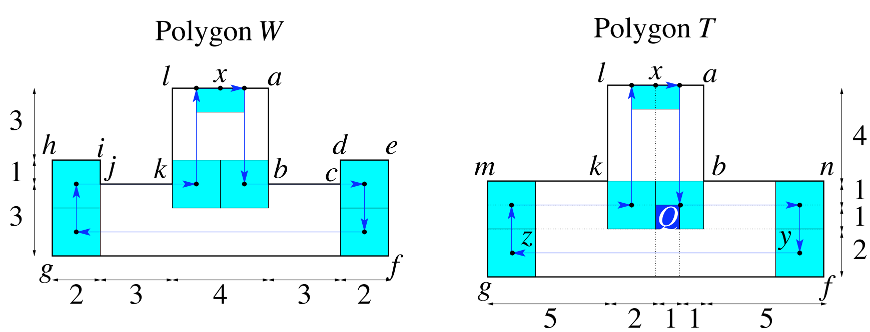

Fig. 2. Optimal solutions in polygon W and T

Lemma 7. *There is neither an optimal on-line algorithm for the exploration, nor for the boundary exploration of rectilinear polygons with:*

1. *a square SA, in the L_1-metric, starting at an arbitrary point of the polygon.*
2. *a round SA, in the L_1-metric, even with the starting point at the boundary.*
3. *a square or a round SA, in the L_2-metric, even starting at the boundary.*

Theorem 1 and Lemmas 6, 7 imply the following result that completely solves the optimality problem of on-line exploration of rectilinear polygons.

Theorem 2. *The only case where on-line exploration of rectilinear polygons can be optimal is the case of the boundary exploration with square SA in the* L_1*-metric, starting at the boundary. Algorithm BOE is optimal in this case.*

4 Competitiveness

Algorithm BOE can be modified to produce a competitive on-line exploration algorithm under all scenarios with square SA.

Theorem 3. *There exists a competitive on-line algorithm for exploration and for boundary exploration of rectilinear polygons with square SA for both metrics and for any starting point.*

We finally turn attention to exploration of arbitrary convex polygons. We present a competitive exploration algorithm, called Algorithm `Convex`, working for arbitrary convex polygons, for round or square SA and regardless of the starting point. We use the L_2-metric, and the result holds for the L_1-metric as well by changing the competitive constant.

The idea of Algorithm `Convex` is the following. First move along a direction until a boundary point becomes explored. Call this distance δ. This is safe, as the optimal algorithm must travel at least the distance $\delta/\sqrt{2}$. Then move along boundaries of increasing squares centered at the starting point, of sizes 2δ, 4δ, 8δ, and so on, until the entire polygon is explored, or until the size of the square is at least 1. (If the boundary of the polygon to be explored prevents the robot from continuing on the square, then it "slides" on the boundary, returning to the travel on the square when again possible.) Since sizes of squares are doubled at each stage, the total trajectory length is at most the double of traversing the last square. If the whole polygon has been already explored, then the trajectory length is proportional to that of the optimal algorithm. Otherwise, the optimal trajectory length is proportional to the diameter and both these values must be at least 1/4. The trajectory length up to this moment is constant, hence making the tour of the polygon boundary and then applying the optimal off-line algorithm to explore its interior (at this point the polygon is known) is proportional to the diameter and hence competitive.

Theorem 4. *Algorithm* `Convex` *is a competitive algorithm to explore any convex polygon, starting from any point, for round or square SA, and for the* L_1 *or the* L_2*-metric.*

5 Conclusion

For the problem of optimality of on-line exploration of rectilinear polygons, our results explain the situation in each of the considered scenarios: we gave an optimal boundary exploration algorithm for a robot with square sensing area starting at the boundary, in the Manhattan metric, while in all other scenarios (exploration of the entire polygon, or arbitrary starting point, or round SA, or the Euclidean metric) we proved that optimal on-line exploration is impossible.

For the problem of competitiveness of on-line exploration of rectilinear polygons, our results are less complete: we showed a competitive algorithm for a robot with square sensing area, regardless of the metric and of the starting point. It is natural to ask if the same result is true for a round sensing area. We conjecture that the answer to this question is positive. It should be noted that competitiveness for the round SA does not immediately follow from competitiveness for the square SA, because there is no bound on the ratio between the lengths of optimal exploration trajectories in these scenarios.

An even bigger challenge would be to show a competitive on-line exploration algorithm for arbitrary polygons, for both shapes of the sensing area. Our competitive algorithm for convex polygons is a step in this direction.

References

1. S. Albers, K. Kursawe, and S. Schuierer. Exploring unknown environments with obstacles. *Algorithmica*, 32(1):123–143, 2002.
2. E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Comput. Geom. Theory Appl.*, 17(1-2):25–50, 2000.
3. W. Chin and S. Ntafos. Optimum watchman routes. In *Proc. of Symposium on Computational Geometry (SCG 1986)*, 24–33, 1986.
4. X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment. In *Proc. of Foundations of Computer Science (FOCS 1991)*, 298–303.
5. X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment: the rectilinear case. *J. ACM*, 45(2):215–245, 1998.
6. Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. In *Int. Conf. of Robotics and Automaton (ICRA 2001)*, 1927–1933.
7. S. Ghosh, J. Burdick, A. Bhattacharya, and S. Sarkar. Online algorithms with discrete visibility - exploring unknown polygonal environments. *Robotics & Automation Magazine*, 15(2):67–76, 2008.
8. M. Hammar, B. J. Nilsson, and S. Schuierer. Improved exploration of rectilinear polygons. *Nordic J. of Computing*, 9(1):32–53, 2002.
9. M. Held. *On the computational geometry of pocket machining.* Springer-Verlag New York, Inc., 1991.
10. F. Hoffmann, C. Icking, R. Klein, and K. Kriegel. The polygon exploration problem. *SIAM J. Comput.*, 31(2):577–600, 2001.
11. C. Icking, T. Kamphans, R. Klein, and E. Langetepe. On the competitive complexity of navigation tasks. In *Revised Papers from the International Workshop on Sensor Based Intelligent Robots*, 245–258, London, UK, 2002. Springer-Verlag.
12. C. Icking, T. Kamphans, R. Klein, and E. Langetepe. Exploring simple grid polygons. In *In 11th Internat. Comput. Combin. Conf*, 524-533, 2005.
13. J. M. Kleinberg. On-line search in a simple polygon. In *Proc . ACM-SIAM Symposium on Discrete Algorithms (SODA 1994)*, 8–15, 1994.
14. B. M. E. Moret, M. Collins, J. Saia, and L. Yu. The ice rink problem. In *Proc. of the 1st Workshop on Algorithm Engineering*, 1997.
15. S. Ntafos. Watchman routes under limited visibility. *Comput. Geom. Theory Appl.*, 1(3):149–170, 1992.