HOW TO LEARN AN UNKNOWN ENVIRONMENT

(Extended Abstract)

Xiaotie Deng[†,†]
[†]Dept of Computer Science
York University
North York, Ont M3J 1P3

Tiko Kameda[†]
[†] School of Computing Science
Simon Fraser University
Burnaby, BC V5A 1S6

Christos Papadimitriou[‡]
[‡]Dept of CSE
UCSD
La Jolla, CA92093

ABSTRACT: *We consider the problem faced by a newborn that must explore and learn an unknown room with obstacles in it. We seek algorithms that achieve a bounded ratio of the worst–case distance traversed in order to see all visible points of the environment (thus "creating a map"), divided by the optimum distance needed to verify the map, if we had it in the beginning. The situation is complicated by the fact that the latter "off-line" problem (the problem of optimally verifying a map) is NP-hard, and thus must be solved approximately. Although we show that there is no such "competitive" algorithm for general obstacle courses, we give a competitive algorithm for the case of a polygonal room with a bounded number of obstacles in it.*

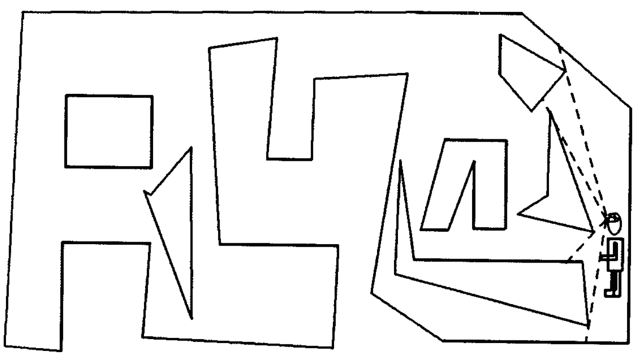


Figure 1: An environment.

1. INTRODUCTION

Imagine that a newborn is in a room with walls and obstacles of unknown, complicated geometry (as an example, see Figure 1). The baby can move around and see the visible portions of the environment. Once a feature is seen, it is memorized for ever, as if a map were drawn during this exploration. The baby must learn her environment; by this we mean that all points in the perimeter of the walls and of the obstacles must be visible from some point in the path traversed. Furthermore, the environment must be learned in an efficient way. Since the exact geometry of the room is unknown, we have to define efficiency in the style of *on–line problems*, as the ratio of the expended effort (e.g., distance traversed) divided by the optimum amount of effort, had we known the room in advance. We are interested in strategies that optimize this ratio. When the ratio is bounded by a constant, the strategy is called *competitive* [ST, KMRS].

Such on–line problems in a geometric context have been considered in the past [PY, BRS], where path lengths are minimized in the on–line sense. The exploration–learning objective of the present paper has also been considered in the literature (see, for example, [Ma] and the references in [BRS]), but not as an on–line problem.

Unfortunately, we can immediately show that general environments are not learnable (in the sense that there is no competitive exploration strategy that works in the general case):

Theorem 1. There is no randomized competitive strategy (against oblivious adversaries) for the case of a polygon with arbitrary polygonal obstacles, even if all polygons are parallelograms.

Sketch: The result follows from a modification of the construction in [PY], by replacing line segments with long, thin diamonds. The randomized case can be established using a techniques similar to those in [KRR]. Notice that non–rectilinear obstacles must be used. □

The main result in this paper is a competitive strategy in the case of an arbitrary polygonal room, with a bounded number of polygonal obstacles in it.

Thus, the present paper can be considered as a

[†] Research supported by the NSERC.
[‡] Research supported by the NSF.

geometric continuation of our previous work on exploring an *unknown directed graph* [DP]; in fact, it addresses and solves the main open problem discussed in that paper[1]. In [DP] we showed that general directed graphs are not learnable, while Eulerian or "almost Eulerian" directed graphs are learnable. However, there are several new difficulties in the present geometric problem, when compared with the graph–theoretic problem studied in [DP]. First, the off–line problem in [DP] (the so–called "Chinese postman problem") is polynomially solvable, providing a sound basis for comparison; whereas the off–line geometric problem ("Given a map, find the shortest path that verifies it visually") can be easily seen to be NP-hard. (Proof: Reduce the Euclidean traveling salesman problem to it by simulating each city by a complex group of blocks that must be visited to be verified...) Even in the case in which the environment is the interior of a simple polygon, there is no known polynomial–time algorithm for the map–verification problem[2]. Hence, we must devise *on–line approximation algorithms for hard problems*, an unfamiliar complication in on–line algorithms, where the off–line problems are usually trivial. More precisely, we must develop approximation algorithms that can be implemented on–line, whose decisions are based only on the features of the environment known at the time the decision is made. An extra complication is that, because of the geometric setting, the choice space is continuous (and it must be made finite by involved canonicalizations and approximations).

We assume that there are two known points in the scene: The *entry*, and the *exit*. We are required to start from the entry, see all parts of the perimeter of the environment, and leave at the exit. (The case in which we are not required to return to an exit can be treated similarly.) It turns out that the off–line problem in the special case where the entrance is the same as the exit, known as the *watchman's route problem*, is polynomially solvable [CN1, CN2]. And it is easy to see that the solution to of the watchman's route problem is at most twice as long as the optimum map verification route. Moreover, solutions to the watchman's route problem have a useful structure: The watchman visits the corners of a particular polygon in the clockwise order.

Still, when the polygon is unknown, we are not able to decide where to start and which direction to follow. We will have to pay an extra price in our ratio for this difficulty. Another useful fact is that if there is an α-competitive algorithm under L_1 metric, then there is a $\sqrt{2}\alpha$-competitive algorithm under L_2 metric. Therefore we can focus on strategies that are competitive for the L_1 metric; and we show that the optimal tours under the L_1 metric are even more restricted.

These techniques enable us to solve the problem only for the interior of a rectilinear polygon (Section 2). More work is needed for the exterior of a polygon, and for the simultaneous presence of walls and obstacles (this is done in the end of Section 2). Technically, the most difficult part of our work is in extending our approach to arbitrary polygons. This is done through a complicated "quantization" technique, explained in Section 3; the competitive ratio we can prove is in the thousands... Finally, in Section 4, we prove lower bounds for the competitive ratio in all these cases, and conclude with open problems.

2. THE CASE OF RECTILINEAR POLYGONS

Now consider a scene that consists of the interior of a simple polygon (not necessarily rectilinear yet). We wish to study paths that see all parts of the interior perimeter of the polygon. The following simple observation is helpful.

Lemma 1. [CN1, Sh] A path sees all parts of the interior perimeter of a polygon iff it sees all sides of the polygon iff it sees all vertices of the polygon. □

For each side s of the polygon P, we extend s inside P (possibly from both sides) until it hits the boundary of the polygon. We thus form a collection of *extended line segments*. That is, each edge of P contributes zero, one, or two such segments. Given the entry (the initial position of the baby in the polygon) and the extension of a side s, the extension may have to be crossed in order to see s. If this is the case, we say that the segment is *necessary*. Obviously, all necessary segments must be "touched" if we are to learn the polygon. A necessary line segment e_1 *dominates* another necessary line segment e_2 if there is no way to cross e_1 without also crossing e_2. Thus, we can drop e_2 from our set of necessary segments. Obviously, dominance can be computed in polynomial time. Furthermore, it can be shown that

[1] In reality, the geometric problem has been the first one we tried, many years ago; our source of inspiration for both problems has been the "critter problem" posed by Ron Rivest in the early 1980's.

[2] We conjecture that it is NP–complete; in the full paper we shall include a nontrivial polynomial algorithm for rectilinear polygons in the L_1 metric.

any sequence of such eliminations elimination steps leaves the same set of segments, which we call the *essential segments*.

Now for *rectilinear* polygons, it is easy to see that each essential line segment intersects at most two essential segments. Moreover, when an essential segment intersects two other essential segments, the entry is between these two latter segments. It follows that for any essential segment e we can cross all other essential segments without crossing e, and we can see all other sides of the polygon but the one extended to generate e, without crossing e.

We can obtain a new polygon P' from P as follows. For each essential line segment e we omit from P the part separated by e that does not contain the entry (se Figure 2). Our task now is simply to visit all essential line segments on the boundary of polygon P'. Unfortunately, when we have to do this on-line, it is impossible to decide where the essential line segments are, and it is even more difficult to decide which line segment to visit first. The following lemma, however, plays a crucial role in our analysis: Consider the minimum-length route (in the L_1 metric) starting and ending at the entry x_0 and visiting the essential line segments of P' in the clockwise order $e_i, e_{i+1}, \ldots, e_m, e_1, \ldots, e_{i-1}$; we call these m routes the *standard clockwise routes*. We can show:

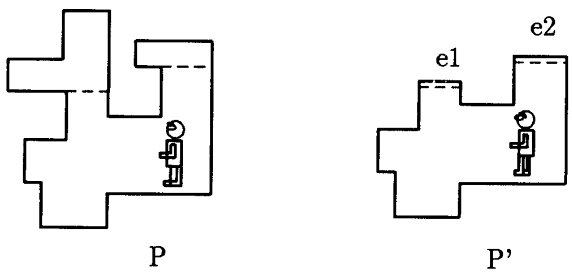


Figure 2: The Polygon P'.

Lemma 2. Each standard clockwise route is at most twice the optimum watchman route. □

Furthermore, it turns out that there is a standard clockwise route which can be obtained by the following greedy algorithm (the entry is denoted x_0):

> for $i = 1$ to m, find the shortest (under L_1) path P_i from x_{i-1} to the line segment e_i and denote the new endpoint of P_i by x_i;
> find the shortest (under L_1) path P_0 from x_m to x_0;

Lemma 3. The greedy algorithm obtains the standard clockwise route that starts from e_1. □

Needless to say, the algorithm works when we start from any segment e_i, instead of e_1. Lemma 3 follows immediately from the following key property of the L_1 metric:

Lemma 4. Let a be a point in a rectilinear polygon, and e a line segment inside the polygon. Suppose a minimum path π from a to e reaches e at point $b \in e$. Then for any other point $c \in e$, the length of any path from a to c will be at least the length of π plus the length of (b, c). □

We design now a strategy that creates a path which is guaranteed to be one of the standard clockwise routes (more precisely, to have length equal to one of them). To simplify matters, we first assume the entry is at the boundary of the polygon P. Thus, there is no choice of a starting e_i: It is whichever contains x_0, call it e_1. We shall extend this to the general case later. Also, it is easy to see that the clockwise order of the essential line segments along the boundary of P is the same as that of P'.

Our strategy maintains a current map M of the polygon, containing all features that have been seen so far. This map will in general consist of several disconnected parts of the boundary of the polygon P. We denote by C the part that contains x_0, and by f (the "frontier") its end in the clockwise direction (see Figure 3). f lies on the line segment $l(f)$. As C expands, f will move in the clockwise direction, but not necessarily continuously: It may "jump" when other fragments of the boundary are merged into C.

Our strategy is the following:

Initially the current position is $x = x_0$, and the map M and C and f consist of the portions visible from x_0. Repeat the following step, choosing among the three principal cases below, until M is completed (and thus f is not well–defined). When this happens, follow the shortest path from x to x_0 and exit. Update M, C, and f whenever a change occurs (say, a line defined by two points of the scene is crossed).

The three cases are these:

(1) *f can be seen from x, and is at a 270° corner (Figure 3(a)); then we move perpendicularly to $l(f)$ towards $l(f)$, until we arrive at $l(f)$ or its extension. If another part of the boundary is encountered, we move parallel to $l(f)$ towards f,*

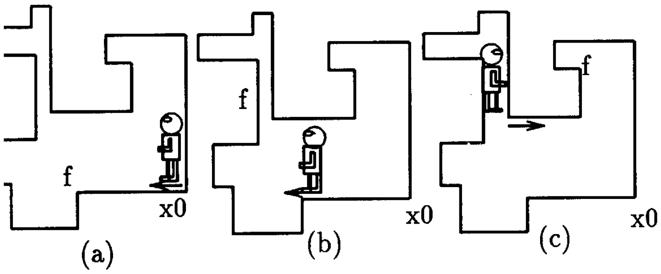


Figure 3: The Strategy.

as necessary. This will always possible, because we can see f from x.

(2) f can be seen from x, and is at an interior point of the line segment $l(f)$ (Figure 3(b)); we do the same as above, until f moves to the end of $l(f)$.

(3) f cannot be seen from x (Figure 3(c)); in this case, f must have just "jumped." We compute the shortest path from x to f (in the L_1 metric) on which f becomes visible as early as possible, and follow it until f is visible. It turns out that we always have enough information to compute this path.

We shall show that this simple algorithm results in a path that is equal in L_1 length to a standard clockwise route. The proof is by induction on the step of the greedy algorithm that creates this route (Lemma 3). Recall the intermediate stops x_i, $i = 0, \ldots, m$ of that algorithm. The induction step is the following:

Lemma 5. Suppose that, at some point, $x = x_{i-1}$. Then, the above algorithm follows either a shortest path (under L_1 metric) from x_{i-1} to e_i, or a shortest path from x_{i-1} to the intersection point of e_i and e_{i+1}, which is x_{i+1}. □

Theorem 2. There is a $2\sqrt{2}$-competitive algorithm for exploring the interiors of rectilinear polygons (in the L_2 metric).

Proof: Follows from the five lemmas, and the comparison between the L_1 and L_2 metrics. □

Finally, if the entry is an interior point, it is difficult to find a standard clockwise route dynamically. However, we may take any one of the four directions along the coordinates from x_0 as the initial direction of motion until the boundary is hit, and then follow a strategy quite similar to the previous one.

Exploring the *exterior* of a rectilinear polygon can be done similarly, once we decide to move around the polygon clockwisely. The technique and lemmas will then be essentially the same, and so is the strategy. There is, however, one difference: We may not have the complete map of the polygon when we reach e_m; we may not even know we have reached the last essential segment (Figure 4). Nevertheless, we can still continue using the same exploring strategy to find the shortest path from x_m to x_0. The details are omitted.

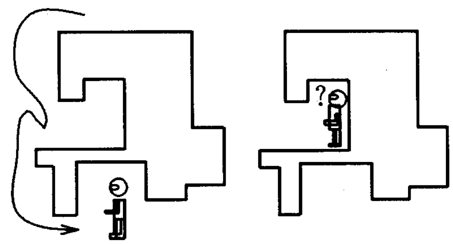


Figure 4: Exploring the Exterior.

Theorem 3. There is a $2\sqrt{2}$-competitive algorithm for exploring exteriors of rectilinear polygons. □

This result, combined with Theorem 2, gives rise to an $O(k)$-competitive algorithm for rectilinear polygons with at most k rectilinear obstacles. The only difficulty is in deciding the order with which to explore the exteriors of the obstacles. Since our algorithm is led by frontier, and Theorem 2, as well as Theorem 3, both hold no matter what order (clockwise or counter-clockwise) we choose, this can be easily dealt with.

Corollary. There is an $O(k)$-competitive algorithm for exploring the interior of a rectilinear polygon with at most k rectilinear obstacles. □

3. GENERAL POLYGONS

Devising a competitive strategy for the interior of a general polygon is much more complicated (approximating the edges by broken rectilinear lines fails to work for several reasons). To start, in the general case, the greedy off-line solution (find the shortest path to the next essential segment) is not guaranteed to produce an optimal watchman's tour. The reason is that the "landing point" x_i can be a bad choice for the next segment (see Figure 5, where B is the greedy choice for x_i, while C is the optimum choice). However, if the essential segments are disjoint, Figure 5 is in some sense the worst that can

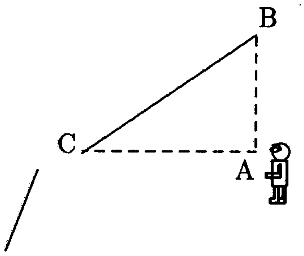


Figure 5: The Greedy Solution is Suboptimal.

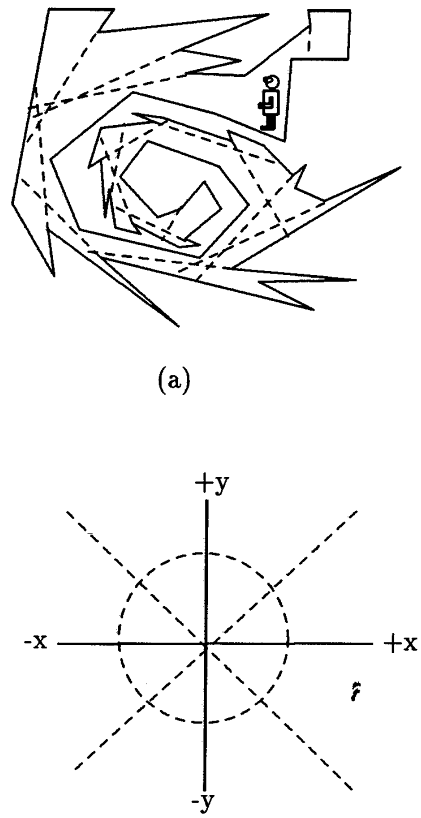
happen (notice that in Figure 5 $AB \leq AC$). Denote the length of the greedy solution G, and the shortest solution S. We can show the following:

Lemma 6. If the essential segments are disjoint, then $G \leq 3 \cdot S$. □

This is important, because it turns out that, like in the rectilinear case, the greedy solution can be implemented by an on–line algorithm. Thus, the difficult case is when essential segments intersect. In this case, consecutively intersecting essential segments can form long groups called *corners* (see Figure 6(a), for example, where only essential segments of category $+y$ are left). It is not difficult to see that the slopes of these segments (say, in the clockwise direction) are strictly increasing in each corner, although they may turn around several multiples of 360° at each corner (forming "spirals"). We subdivide all possible slopes into eight octants, and into four categories: $+x$, $-x$, $+y$, and $-y$ (see Figure 6(b)). We also say that categories $+x$ and $-x$ form a *larger category* called $\pm x$, and similarly for y. Each corner is then subdivided into several consecutive groups called *intervals*, where an interval contains consecutive segments in the same octant (there are possibly more than eight intervals per corner, if the corner is a spiral). Suppose now that we run the greedy algorithm, except that we do not visit all essential segments, but *the last segment in each interval*. Call the length of the resulting route G^1. We can show the following, by a case analysis:

Lemma 7. $G \leq 3 \cdot G^1$. □

Suppose now that the two larger categories $\pm x$ and $\pm y$ of the segments traversed in G^1 are visited separately by a greedy algorithm. The corresponding lengths are denoted $G^{\pm x}$ and $G^{\pm y}$. The following result is perhaps the key (and most difficult) step in our proof:



(a)

(b)

Figure 6: Corners and Intervals.

Lemma 8. $G^1 \leq 16(G^{\pm x} + G^{\pm y})$. □

Now, if we define analogously the lengths G^{+x}, $G+-x$, etc., for the refinements corresponding to the smaller categories, an interesting potential function argument establishes the following inequality (and similarly for y):

Lemma 9. $G^{\pm x} \leq 7(G^{+x} + G^{-x})$. □

However, it turns out that the essential segments in the category $+x$, for example, are "almost" disjoint, and thus Lemma 6 applies to bound G^{+x} etc. from above by $3 \cdot S$. Combining the four lemmas, we have that $G \leq 2016 \cdot S$. Since we can implement the greedy algorithm in an on–line fashion, similar to the frontier–lead strategy in the rectilinear case, we have:

Theorem 4. There is a competitive algorithm for exploring interiors of general polygons. □

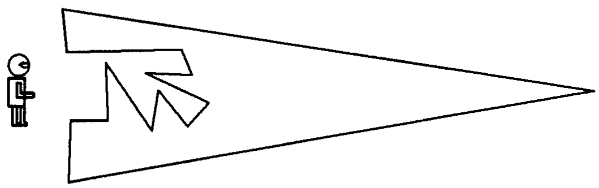


Figure 7: Exploring the Exterior.

It is nontrivial to extend this result to exteriors of general polygons. The main difficulty is exemplified in Figure 7, where the optimum exploration of the exterior does not traverse any of the long edges. To obtain a competitive algorithm we need to apply ideas from the "layered graph traversal" algorithm from [PY]. Finally, we have our main result:

Theorem 5. There exists a competitive algorithm for exploring interiors of general polygons which has a bounded number of polygonal obstacles. □

4. LOWER BOUNDS AND OPEN PROBLEMS

In the introduction, we have given a lower bound for general polygons. Here, we bound from below the achievable competitive ratios in rectilinear polygons:

Theorem 6. There is no deterministic strategy that has a competitive ratio less than

(a) $\frac{1+\sqrt{2}}{2}$, for exploring the interior of an unknown rectilinear polygon;

(b) $\frac{1+\sqrt{2}}{2}$ for exploring the exterior of an unknown rectilinear polygon;

(c) $\sqrt{2}$ for exploring an unknown rectilinear polygon with rectilinear obstacles. □

When we have more that one obstacles, it is open whether there is a lower bound which is an increasing function of the number of obstacles. The lower bound of $\sqrt{2}$ holds for a single obstacle. *We know of no better bounds when the number of obstacles is unbounded.*

The obvious open question is to bridge the gap between the upper and lower bounds. Especially pronounced are the gaps in the more complicated case of general polygons. We conjecture that a more reasonable competitive ratio (such as three) is possible. But a more intriguing question is, whether or not there is a competitive algorithm for an unbounded number of rectilinear obstacles. We conjecture that such a competitive algorithm is possible.

Finally, by a standard construction, we know that there is no competitive algorithm for exploring 3-dimensional polyhedra, even without obstacles. However, the problem for rectilinear polyhedra remains open.

References.

[BRS] A. Blum, P. Raghavan and B. Schieber, "Navigating in Unfamiliar Geometric Terrain," *STOC 1991.*

[CN1] W. Chin, and S. Ntafos, "Optimum Watchman Routes," *Information Processing Letters* **28**, pp. 39-44, 1988.

[CN2] W. Chin, and S. Ntafos, "Shortest Watchman Routes in Simple Polygons," to appear in *Discrete and Computational Geometry.*

[DP] X. Deng, and C. H. Papadimitriou, "Exploring an Unknown Graph," *Proc. of the 31st Annual Symposium on Foundations of Computer Science*, 1990, pp.355-361.

[JP] D. S. Johnson, and C. H. Papadimitriou, "Computational Complexity," in *The Travelling Salesman Problem*, edited by E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys, pp. 37-85.

[KRR] H. Karloff, Y. Rabani, and Y. Ravid, "Lower Bounds for Randomized *k*-Server and Motion-Planning Algorithms," *STOC 1991.*

[Ma] Maja J. Mataric, "A Distributed Model for Mobile Robot Environment-Learning and Navigation," *Tech Report #1228, MIT AI Lab*, May 1990.

[NG] S. Ntafos, and L. Gewali, "External Watchman Routes," manuscript, 1989, presented in *First Canadian Conference on Computational Geometry.*

[O'R] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, Oxford, 1987.

[PY] C. Papadimitriou and M. Yanakakis, "Shortest paths without a map," *Proc. 16th ICALP*, 1989, pp.610-620. To appear in the special issue of *Theoretical Computer Science*, 1991.

[Sh] T. Shermer, "Recent Results in Art Galleries," *CMPT TR90-10, School of Computing Science, Simon Fraser University*, Oct., 1990.

[ST] Sleator, D.D. and Tarjan, R.E., Amortized efficiency of list update and paging rules," *Communications of the ACM 28(2)*, 1985, pp.202-208.