SERIE B — INFORMATIK

An Efficient Competitive Strategy for Learning a Polygon

Frank Hoffmann * Christian Icking **Rolf Klein **
Klaus Kriegel *

B 96–01
February 1996

Abstract

We provide a competitive strategy for a mobile robot with vision, that has to explore an unknown simple polygon starting from and returning to a given point x_o on the boundary. Our strategy creates a tour that does not exceed in length 133 times the length of the shortest watchman route from x_0. It has been claimed before by other authors that a competitive strategy with factor 2016 exists for this problem; but no proof has appeared except for the easy rectilinear case.

*Freie Universität Berlin, Institut für Informatik, Takustr. 9, 14195 Berlin, Germany.
hoffmann@inf.fu-berlin.de, kriegel@inf.fu-berlin.de

**FernUniversität Hagen, Praktische Informatik VI, Elberfelder Str. 95, 58084 Hagen, Germany.
Rolf.Klein@FernUni-Hagen.de, Christian.Icking@FernUni-Hagen.de

1 Introduction

Two basic tasks of autonomous mobile robots are to search for a goal in an unknown environment, and to explore an unknown environment, i. e. to walk around until each point of the environment has been visible at least once.

Such problems have a long history; Shannon's mouse [25] has been searching labyrinths, and the Pledge algorithm, as described in Abelson and diSessa [2] and independently discovered by Asser [1] was the first correct solution to the general problem of escaping from a labyrinth, where the robot is equipped only with a touch sensor and a turn counter. In [21] and numerous subsequent papers Lumelsky and Stepanov have studied a robot that knows coordinates. They were the first to provide some sort of upper bound to the length of the robot's path, in terms of properties of the environment.

A sharper way of measuring performance was introduced to computational geometry by Baeza-Yates et al. [3]. They presented strategies for finding a goal in an unknown environment that are *competitive* in the sense that the length of the path from the start to the goal created by the strategy does not exceed the length of the shortest path, times a constant competitive factor.

This concept has been known long before in other areas; examples are the first fit strategy for solving the bin packing problem [11], or self-organizing data structures as studied by Sleator and Tarjan [24]. A survey on competitive strategies can be found in Ottmann et al. [22], strategies for robot navigation have been reviewed in e. g. Rao et al. [23], and in Icking and Klein [15].

Usually one assumes that the robot's environment is a (simple) polygon. The robot is equipped with a 360° vision system that provides the visibility polygon in real time. The cost of moving dominates the cost of computing; it is often measured by path length, sometimes by the number of turns.

Many results on competitive *search* algorithms have appeared. For example, Blum et al. [4] have studied several problems involving obstacles. Special polygons called *streets* have been investigated by Klein [18], Kleinberg [19], Lopez-Ortiz and Schuierer [20], Datta and Icking [8], and Ghosh and Saluja [12].

Relatively few competitive strategies are known for *learning* an unknown environment. Icking, Klein, and Ma [17] gave an optimal competitive strategy for looking around a single corner. Recently Icking and Klein [16] have shown how to find the closest point in an unknown star-shaped polygon from which the whole polygon is visible. This is related to off-line guard problems; see e. g. Hoffmann [13] and Hoffmann et al. [14].

One of the most interesting problems in this area is how to learn an arbitrary environment in a competitive way. Here we assume that the robot starts from a given point x_0 on the polygon boundary, walks through the polygon, and eventually returns to the start point x_0. On its way, it must see each point of the polygon's boundary. In computing the competitive factor, the length of this closed path is compared with the length of the *shortest* closed path from x_0 from which each point of the boundary is visible. This path is called the *shortest watchman tour* from x_0.

A lot is known about computing the optimal solutions if the polygon is given. Chin and Ntafos [7] have shown that the shortest watchman tour from x_0 can be computed in time $O(n^4)$; their result has later been improved on by Tan and Hirata [26] to $O(n^2)$. Carlsson et al. [6] have shown that the shortest watchman tour without specified point x_0 can be computed in time $O(n^3)$. Recently, Carlsson and Jonsson [5] proposed an $O(n^6)$ algorithm for computing the shortest path inside a simple polygon from which each point of the boundary is visible, where start and end point are not specified. These results do not directly translate into competitive strategies that serve the same purpose in an unknown polygon; however, they provide useful structural information about the nature of the optimal solution.

In Deng et al. [9] it has been claimed that there exists a competitive strategy for learning an unknown polygon with a competitive factor of 2016. A complete proof of this claim has, to our knowledge, not appeared. Only for the case of *rectilinear* polygons has this claim been substantiated in Deng et al. [10]. Rectilinear polygons are quite easy to explore, because one knows exactly which line in the polygon to visit in order to look around a corner. Exploiting this fact leads to a competitive strategy with factor 1 (with respect to L_1–metric).

In this paper, we provide a competitive strategy, called *Scout–and–March*, with factor 133 for learning arbitrary simple polygons.

The paper is organized as follows. In the next section we provide basic notations and introduce the notion of the discovery tree of a simple polygon which helps to describe the order vertices are discovered by a watchman tour.

Section 3 presents the recursive scheme of the Scout–and–March strategy which is based on decomposing the polygon on–line into smaller 'rooms' separated by 'doors'. Each room splits in two uniform parts, one consisting of left vertices the other one of right vertices only. We estimate the length of a Scout–and–March tour assuming a competitive strategy Scout$_1$ for uniform polygons. This basic Scout$_1$ –modul is presented and analyzed in Section 4. We conclude with summarizing our results and discussing related open problems in Section 5.

2 Preliminaries

2.1 Notations and Basic Facts

Let a simple polygon P be given by a list $v_1, v_2, \ldots, v_n$ of its boundary vertices in clockwise order. We assume further that P is in general position, i.e., there is no line with three vertices on it.

The edge $[v_{i-1}, v_i]$ is called the *left edge* of v_i and the edge $[v_i, v_{i+1}]$ is called the *right edge* of v_i.

As usual, P will be understood as the the union of its interior $int(P)$ and the boundary $bd(P)$ of the polygon.

A point $y \in P$ is *visible* from a point $x \in P$ if the closed line segment $[x, y]$ is contained in P. $Vis(x)$ denotes the visibility polygon of x, i.e. the set of all points

$y \in P$ visible from x.

A *watchman path* C is a curve in P such that any point y in P is visible from some point on the curve. Sometimes it is usefull to consider parametrized curves $C : [0,1] \to P$. In this case $C(0)$ is called the start point of the watchman path and if, moreover, $C(0) = C(1)$ then C is called a *watchman tour*.

In the whole paper we assume that a fixed start point $x_0 \in P$ is given.

By $p_P(x,y)$ (or $p(x,y)$ for short) we denote the shortest path in P connecting x with y and let Π_x be the shortest path tree describing all shortest paths from a fixed vertex x to all other vertices of P. The *father node* $f(y)$ of a polygon vertex y is its predecessor in Π_{x_0}. For an arbitrary point $y \in P$ its father $f(y)$ is defined to be the first link point on the shortest path in P from y to x_0 provided that there is a link on this path, otherwise it is x_0.

There is a different characterization of the father vertex $f(y)$ of a point $y \in P$. If y is visible from x_0 then we define $f(y) = x_0$. Otherwise, we consider the the visibility polygon $Vis(y)$. Its boundary consists of polygon edges, segments of polygon edges, and spurious edges separating caves from $Vis(y)$. Since x_0 is not visible from y and P is a simple polygon it is in a uniquely determined cave B. Obviously, there is a reflex vertex v such that the spurious edge separating B from $Vis(y)$ is the extension of the visibility ray from y to v. Then we define $f(y) = v$ and the spurious edge emanating from v is called the *discover line* of y, denoted by $\delta(y)$, see Figure 1.

The following fact is easy to observe: If C is an arbitrary watchman path starting in x_0 and x is the first point on C from which y is visible (y is discovered from x) then x is on $\delta(y)$.

Let v be a polygon vertex and let H_l (resp. H_r) be the inner halfplane of the left (right) edge of v. We subdivide $Vis(v)$ into three polygonal regions: $(H_l \setminus H_r) \cap Vis(v)$, $(H_r \setminus H_l) \cap Vis(v)$, and $H_l \cap H_r \cap Vis(v)$. Note that $f(v)$ is in exactly one of these regions. According to this region we label vertex v either $left$, $right$, or $neutral$.

Figure 1 illustrates this definition: With respect to start point x the label of v is *right*. With x' as start point v would have label *neutral*. Note that a non–reflex vertex is always labeled $neutral$. A *left* (resp. *right) polygon* is a polygon with start point such that all its reflex vertices are left (right) or neutral.

If u is a descendant of v in Π_{x_0} we write $u < v$. Let v be a left (right) vertex. There is a spurious edge $\sigma(v)$ of $Vis(f(v))$ which is incident with v. This spurious edge will be called the *shadow line* of v because it hides a 'shadowed' cave $S(v)$ which is not visible in the moment when v gets discovered and hasn't been visible before. Obviously, the shadow $S(v)$ contains the right (left) edge of v as well as all descendants $u < v$. We remark that both the discover line $\delta(v)$ and the shadow line $\sigma(v)$ are extensions of the visibility line segment $[v, f(v)]$. Note that neutral vertices and especially convex vertices have no shadow.

Let $C : [0,1] \to P$ represent a watchman path. We say that a vertex v is discovered by C at time t if $C(t)$ is the first point on C visible from v, i.e. C touches $\delta(v)$ the first time.

Let v be a left (right) vertex and v' the next polygon vertex in clockwise (coun-

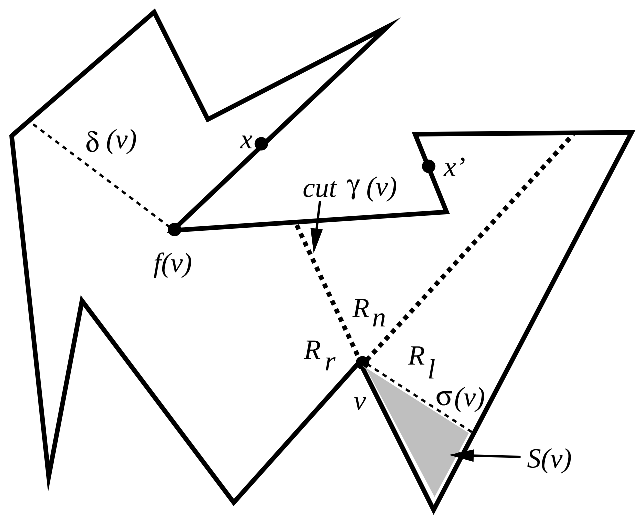


Figure 1: Discover line, shadow, father, and label of vertex v.

terclockwise) order. Then we say that v has been *explored* by C at time t iff t is the discover time of v'.

So the usual situation is: In the beginning a vertex is hidden by its father, then it gets discovered and we know its label, and eventually it is explored. More precisely we have the following simple basic fact.

Proposition 2.1. A curve C starting in x_0 is a watchman path iff it explores all left and all right vertices.

□

Let in the following W always denote the unique clockwise oriented optimal watchman tour for P with start point $x_0 \in bd(P)$. Further denote the finite area encircled by W on its right side by $[W]$. Recall that a region R within a polygon P is relatively convex if for any points $x, y \in R$ we have $p(x, y) \subset R$. The following is a well–known property of optimal watchman tours.

Proposition 2.2. $[W]$ is relatively convex in P.

□

Especially, for points $x, y \in W$ we know that every link point of $p(x, y)$ is a link point of W.

2.2 Transit Vertices

Of crucial importance throughout the paper are those left or right vertices which have sons of opposite label. For a left or right vertex v we denote by $s(v)$ that son with different label discovered first. If no such son exists, $s(v)$ is undefined. For the

start point x_0 we set $s(x_0) = x_0$ and we will treat it like a right vertex.
Denote by T the set of all vertices v for which $s(v)$ is defined. We call these vertices *transit vertices* because of the following simple consequence of Proposition 2.2.

Proposition 2.3. It holds $T \subset W$.
Proof. Assume there is some $v \in T \setminus W$. To see a point from the interior of the shadow $S(s(v))$ the tour W has to cross the visibility segment $[v, s(v)]$, say in some point w different from v. But $f(w) = v$ and we have a contradiction to Proposition 2.2.

□

To be precise we remark that there can be other vertices than those from T the optimal tour passes through. Vertices from T however can be easily recognized by an on–line strategy.

In Figure 1 vertex v is a right vertex, its father $f(v)$ a left vertex. Hence the optimal watchman tour starting in x goes through $f(v)$.

2.3 Watching Subpolygons

Next we analyze which parts of the optimal watchman tour W suffice to watch certain relatively convex subpolygons of P.

Let X be a set of at least two polygon vertices. Assume that $X \subset W$. Consider a polygon P' obtained from X by either connecting two consecutive (with respect to their circular order on W) vertices $x, x' \in X$ either by the connecting diagonal (assuming it to be contained in P) or by the corresponding subpath of $bd(P)$. For each subpolygon P' defined this way we have the following.

Proposition 2.4. P' is watched from $P' \cap W$.
Proof. By contradiction, assume that there is a point $z \in int(P')$ not seen from $W \cap P'$. On the other hand, z is watched from some $w \in W \setminus P'$ implying that the visibility segment $[z, w]$ cuts a diagonal $[x, y]$ on the boundary of P'. We can assume that no other point on the visibility segment $[z, w]$ is from W. From Proposition 2.2 we have that $[z, w] \subset [W]$.The line through z and w must cut W a second time, say in w', since W is a tour. But w' is not on the same side of z, therefore $[z, w']$ cuts a second diagonal $[x', y'] \subset bd(P')$. Now we can use the following simple basic fact about watchman tours in polygons. It says, that for an arbitrary point in a polygon its range of visibility contact with the tour (regarded as a subset of the unit sphere) is a connected arc. From this we know that there is a possibility to turn the visibility ray from $[z, w]$ to $[z, w']$ without loosing visibility contact to W and we are done.

□

It is essential in the previous statement that all vertices in X are on the optimal tour, compare also with the example in Figure 4.

Next we study the situation when not all vertices from X are on the optimal tour. Let $[x, y]$ be a polygon diagonal which is part of $bd(P')$ and assume that not both x, y are on W. Say W is oriented such that it leaves P' through $[x, y]$ in a and reenters it in some point b. If W is not empty behind $[x, y]$ let a^* denote the first link point one reaches traversing on W outside P' from a in direction b and b^* the first link point on W traversing from b to a. Let $(W \cap P')^*$ be the union of $W \cap P'$ and all segments $[a, a^*], [b, b^*]$ for all polygon diagonals on $bd(P')$.

Corollary 2.5. A subpolygon P' defined by diagonals is watched from $(W \cap P')^*$. □

3 The *Scout–and–March* Strategy

3.1 Doors and Rooms

Let P be a polygon, $x_0 \in bd(P)$ the fixed start point and Π_{x_0} the corresponding shortest path tree.

We recursively define *doors* of P as follows. To initialize the definition we say that $x = x_0$ is an *entrance door* with label right. Next we associate with an entrance door x a possibly empty set D_x of *exit doors*. Each $y \in D_x$ then recursively serves as a new entrance.
We define D_x to be the set of all reflex vertices y such that:

1. y is transit vertex,
2. y has a label different from x,
3. on the path in Π_x from x to y there are no other vertices with property (1) and (2).

For a transit vertex $x \in T$ we denote by P_x the subpolygon of P behind (with respect to x_0) the diagonal $[x, s(x)]$ and we set $P_{x_0} = P$.

Let D denote the set of all doors of P with respect to start point x_0. With each $x \in D$ we associate a subpolygon R_x – called the *room with entrance door x* – by

$$R_x = P_x \setminus \bigcup_{y \in D_x} P_y$$

.

Remark 3.1. Obviously, different rooms have disjoint interiors and $P = \bigcup_{x \in D} R_x$. Another important property of a room R_x is that no point from $(x, s(x))$ can see a point from $(y, s(y))$ for each $y \in D_x$. This follows from the fact that entrance and exit doors of a room have different labels.

We describe a room R_x as the union of left and right subpolygons as follows. If x is left we know that the subpolygon

$$R_x^1 = R_x \setminus \bigcup_{u \in R_x \cap V^r} S(u)$$

is a left polygon, where V^r denotes the set of all right polygon vertices. x itself is neutral in R_x^1. Moreover, $R_x^2 = R_x \setminus R_x^1$ is the union of right polygons, since a left vertex in this region would imply a right father and therefore it would be one vertex of an exit door diagonal. For a right door the situation is symmetric.

We illustrate some of these notions by a first example shown in Figure 2. With respect to start point x_0 we have $T = \{x_0, v_1, v_2, v_4, w_1\}$. Observe that $v_3 \notin T$, since its only son w_3 has the same label. Moreover, $D_{x_0} = \{v_1, v_4\}$. The subpolygons $R_{x_0}^1$ and the components of $R_{x_0}^2$ are shaded.

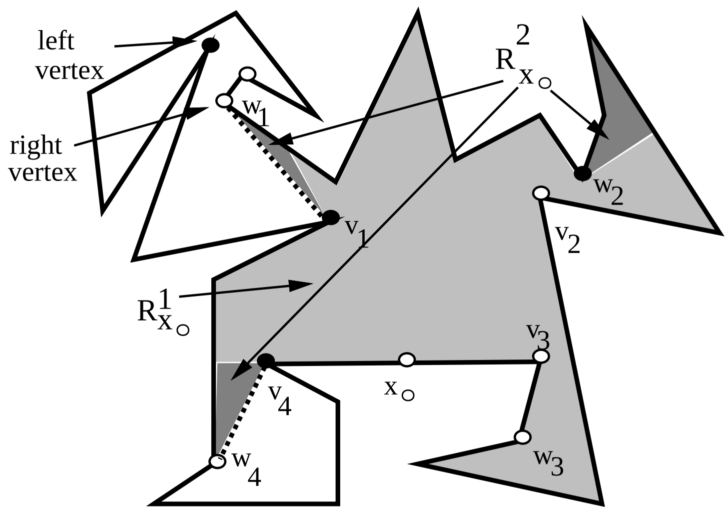


Figure 2: Right and left subpolygon of the room R_{x_0}.

The next example in Figure 3 shows entrance and exit door diagonals of two rooms.

We know that for each $u \in D$, $u \neq x_0$, the tour W crosses the diagonal $[u, s(u)]$ twice, once in u and a second time in a possibly different point which we denote by $b(u)$. More precisely, we know for a left door u that the diagonal $[u, s(u)]$ is the first time reached in u and, eventually, left in $b(u)$. A right door is entered in $b(u)$ and left in u.

Hence we know from Corollary 2.5 which parts of W suffice to watch a room. Let us denote $W \cap R_x$ by W_x and the extension $(W \cap R_x)^*$ by W_x^*. We have

Theorem 3.2. Each point in a room R_x is watched from W_x^*.

□

In the illustrating Figure 4 we observe that for the entrance door x we have $b(x) = b^*(x) = s(x) = y$. In general, however, an on–line strategy cannot recognize before crossing the diagonal $[x, s(x)]$ whether $s(x)$ is possibly a transit point or not. Remark that $b^*(x)$ is defined with respect to a room, so there are two such points associated with a diagonal $[x, s(x)]$.

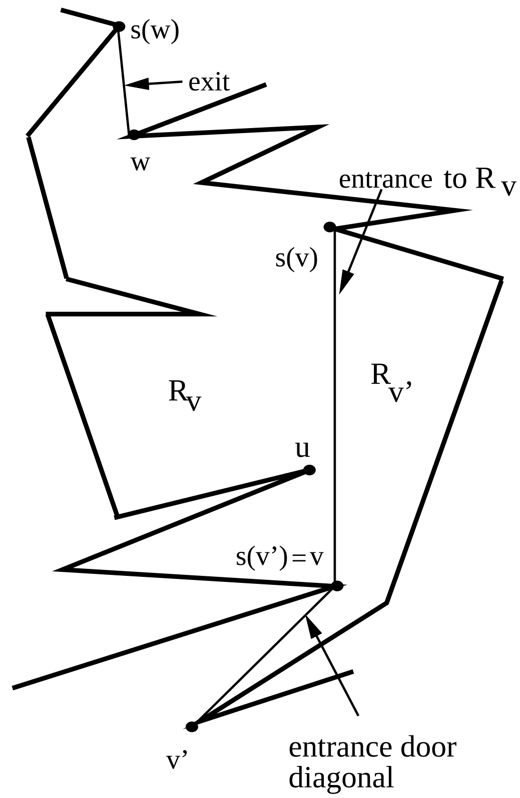


Figure 3: Rooms and doors.

3.2 Outline of the Strategy

The *Scout–and–March* strategy, or SAM for short, works recursively in a Depth–First–Search mode. All rooms to be learned by the strategy are kept as a stack of their entrance doors. In the beginning $x = x_0$ is the only door in the stack. Each step of the recursion has two phases: Firstly, the strategy explores (i.e., "scouts") the room R_x (where x is the current top element of the stack) starting from its entrance door with the aim to learn all exits, i.e., to learn D_x. Then, in a second stage, the strategy marches from x to the first exit of R_x in cyclic order.

Scouting Stage. The scouting stage consists of two exploration tours described in detail in the next section.

1. Scout$_1$: SAM learns the subpolygon R_x^1 ending up again in x. Let E_x^1 be this tour. Then SAM computes the shortest tour S_x^1 that connects x with the different connected components of R_x^2.

2. Scout$_2$: SAM learns R_x^2 using its knowledge about S_x^1 on an exploration tour E_x^2 and recognizes all exits D_x and pushes them in cyclic order on the stack.

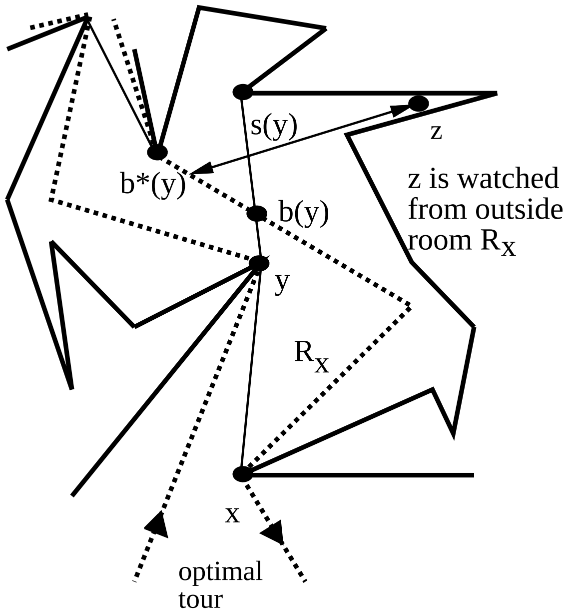


Figure 4: Watching room R_x from W_x^*.

Marching Stage: SAM marches on the shortest path $p(x, y)$ from x to y, the first entrance of a room still to be learned, if there is some. Otherwise SAM returns to the starting point x_0.

Remarks 3.3.

1. Observe that during scouting a room R_x we do not explore vertices outside R_x. That means, that after discovering an exit door y we stop exploring sons of y which have the same label as y and are in P_y. This task SAM performes immediately after entering R_y when it starts scouting R_y by exploring vertices with the same label as y. Of course, there need not be such vertices. For the second phase of the exploration of R_y we certainly know that SAM makes some progress since at least $s(y)$ has to be explored.

2. Not all transit vertices are used by SAM as doors. If v is a left transit vertex it is quite possible to see from points on $(v, s(v))$ points from some $(u, s(u))$ where u is a left vertex from T and at the same time son of v. Vertex u in Figure 3 for example is of that kind.

3.3 The Length of a Scout–and–March Tour

We state two remarks that will help to estimate the length of the SAM–tour.

A. In the next section we prove that there is a competitive ratio c_0 for learning a left/right subpolygon of P using Scout_1 implying a competitive ratio $2c_0$ for learning a room.

B. Let $x \in D$ be an entrance door in P. We know $x \in W$ because $x \in T$. We have $|[x, b(x)]| \le |W_x|$. This easily follows from the following observation. Denote by W_x^0 the image of a parallel projection of $[x, b(x)]$ onto W_x. The total length of W_x^0 (which can be disconnected) is at least $|[x, b(x)]|$ because of Remark 3.1.

Proposition 3.4. The total length of all paths used by SAM during the marching stages is shorter than $|3W|$.

Proof. Consider a situation when W enters a room R_x through a (right) door diagonal in point $b(x)$ (and leaves it, eventually, in $x \in D$) while SAM enters and leaves it in x. Let y be the next exit (a left door) reached both by W after traversing $W(b(x), y)$ and by SAM. Both reach it in y but SAM traversed $p_P(x, y)$. Now

$$|p_P(x, y)| \le |W(b(x), y)| + |[x, b(x)]|.$$

Applying remark B and summing up over all rooms gives the inequality since a fixed part W_x of W is, in worst case, used twice for the estimation inside R_x and once for the predecessor room.

□

Figure 5 gives a schematic view of the situation. The broken lines indicate W_x. The paths marched by SAM connect transit vertices. Therefore they are inside $[W]$.

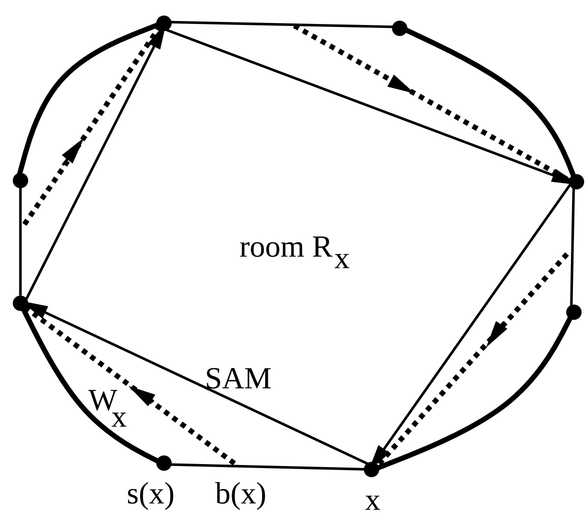


Figure 5: Estimating the marching stage.

Let us next estimate the sum of the lengths of all scouting tours in all rooms generated by SAM.

Proposition 3.5. $\sum_{x \in D}(|E_x^1| + |E_x^2|) \leq 10c_0|W|$.
Proof. Let $x \in D$. We know by Theorem 3.2 that W_x^* watches R_x. In general this is not a closed tour. To get a tour watching R_x we attach to W_x^* for each exit door $y \in D_x$ the path $p(y, b^*(y))$.

The entrance $[x, s(x)]$ is treated separately. Here we add the door diagonal segment $[x, b(x)]$ and the path back from $b^*(x)$ to $b(x)$.

We get a closed tour obviously watching R_x which we denote by W_x^{cl}. Further, consider an optimal local watchman tour W_x^{loc} which starting from x has to learn the room R_x only.

Certainly we have $|W_x^{loc}| \leq |W_x^{cl}|$. On the other side, $|E_x^1| + |E_x^2| \leq 2c_0|W_x^{loc}|$ by Remark A. Finally, we show that $\sum_{x \in D} |W_x^{cl}| \leq 5|W|$ what implies the claim.

To see this consider an exit $y \in D_x$. By the projection argument we have that

$$|p(y, b^*(y))| + |[b(y), b^*(y)]| \leq |W_y|.$$

Here it is important that $b^*(y)$ was chosen to be the first link point behind $b(y)$. Moreover to connect an interior point of a (left) entrance door diagonal with an interior point of a (right) exit diagonal one needs at least one link with a (right) turn. Therefore we can project $p(y, b^*(y))$ onto W_y without hitting an exit diagonal or the segment $[b(y), b^*(y)]$ of W.

With the entrance $[x, s(x)]$ one has to be more careful. The projection argument cannot be applied, since $[x, s(x)]$ is exit of a room, say R_w, which may have many exits. However, recall that for different exits $x, x' \in D_w$ the points $b^*(x), b^*(x')$ are different and that $|[x, b(x)]| \leq |W_x|$.

Counting again how often a fixed W_x is used to estimate the length of some W_y^{cl} gives the inequality. W_x is during the estimation in worst case used:
(1) twice for W_x^{cl},
(2) once for the exit of its predecessor room,
(3) twice for entrance doors of its successor rooms.

□

Theorem 3.6. The length of a *Scout–and–March* tour is bounded by $(10c_0 + 3)|W|$, where c_0 is the competitive factor of Scout_1 to learn left/right subpolygons of P.

□

4 How to Scout a Room

In this section we present a competitive strategy for learning a room R_x. W.l.o.g. we may assume that the entrance door is defined by a right transit vertex x. The left case is symmetric: Exchange left for right and clockwise for counterclockwise. Recall that we need first a competitive strategy Scout_1 for learning the right subpolygon R_x^1 (i.e., the room R_x without the shadows of its left vertices). Scout_1 will report a list of all right transit vertices in R_x. The second procedure Scout$_2$ will generate a shortest tour connecting these transit vertices and in each one it will start an exploration of the corresponding component of $R_x \setminus R_x^1$ which is a left polygon.

Once again, learning a subpolygon $P' \subset P$ means that both our strategy and the optimal watchman tour for P' can move within the whole P while learning P'.

4.1 Looking Around a Corner

We start by describing the very basic modul of how to explore a single right vertex. This is known as the *How to look around a corner*–problem and the best known solution for exploring a visible corner without any further restrictions (i.e., without possible obstacles) is described in [17].

The so–called *halfcircle strategy* HCS introduced below has a competitive ratio, which is a little bit larger. This strategy is, however, simply adaptable to scenes with obstacles.

We start with some notations and two basic elementary facts. Given two points x and y in the plane $C_l(x,y)$ (resp. $C_r(x,y)$) will denote the left (right) halfcircle over the oriented line segment $[x,y]$.

Fact 4.1. Let x be a point on a circle of diameter d and consider two rays emanating from x that cross the circle once more. Then the length of the arc between the two rays is αd where α is the angle between the two rays

□

Fact 4.2. Let x, x', and y be three points which are not colliniar and such that the two halfcircles $C_l(x,y)$ and $C_l(x',y)$ intersect in a point $z \neq y$. Then the points x, x', and z are colliniar.

□

The next fact (illustrated in Figure 6 and Figure 7) is still simple but a bit more complicated to prove.

Fact 4.3. Let $p = x_0, x_1, \ldots, x_n$ be a simple path which turns to the right at each point x_i, $0 < i < n$, by an angle α_i and let Env be the upper envelope of the set of halfcircles $\{C_{i,j} = C_l(x_i, x_j) \mid 0 \leq i < j \leq n\}$.
Then for the lenght $|Env|$ we have

$$|Env| \leq \frac{\pi}{2} \sum_{i=1}^{n} |[x_{i-1}, x_i]|$$

Moreover, if x is a point on Env and x_k is the first point on p visible from x then the length of the initial part of Env from x_0 to x is at most $\frac{\pi}{2}(\sum_{i=1}^{k} |[x_{i-1}, x_i]| + |[x_k, x]|)$.

Proof. W.l.o.g. we may assume that $\alpha_i < \pi/2$ for any $0 < i < n$. Otherwise x_i were on Env and it would be possible to subdivide the problem. Note that for any halfcircle $C_{i,j}$ contributing to Env we have $\alpha_{i+1}+\alpha_{i+2}+\ldots+\alpha_{j-1}<\pi/2$. Moreover for any point $z \in Env$ one can find its halfcircle $C_{i,j}$ by choosing i (resp. j) the minimal (resp. maximal) number such that both x_i, x_j are visible from z. With these notations, Env consists of segments of the following halfcircles only:

$$C_{0,1}, C_{0,2}, \ldots, C_{0,j(0)}, C_{1,j(0)}, C_{1,j(0)+1}, \ldots, C_{1,j(1)}, C_{2,j(1)}, \ldots, C_{j-1,j}.$$

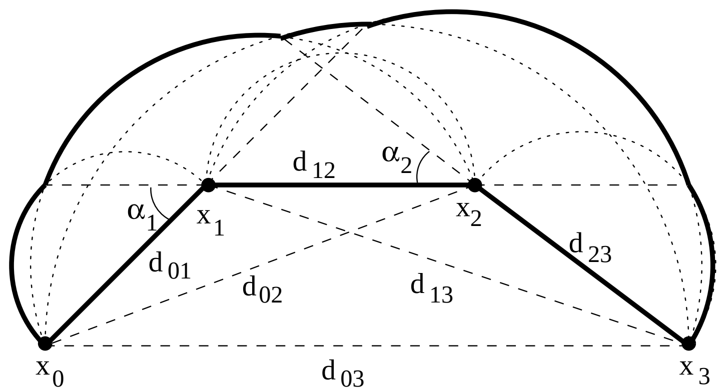


Figure 6: The envelope Env of halfcircles over a path p.

In the following $d_{i,j}$ denotes the Euclidean distance between x_i and x_j. We will scan Env by a clockwise turning ray which is tangent to the path p. We start turning the ray $\overrightarrow{x_1, x_0}$ around x_1 by the angle α_1. The length of the scanned segment of $C_{0,1}$ is $\alpha_1 d_{1,1}$ (Fact 4.1). By Fact 4.2 at the current point on Env the halfcircles $C_{0,1}$ and $C_{0,2}$ intersect. So we turn the ray $\overrightarrow{x_2, x_1}$ around x_2 by the angle α_2 scanning a segment of $C_{0,2}$ of length $\alpha_2 d_{0,2} \leq \alpha_2(d_{0,1}+d_{1,2})$.

We continue analogously until we have to turn the ray $\overrightarrow{x_{j(0)}, x_{j(0)-1}}$. This step splits into two parts:
First we turn this ray by the angle $\alpha'_{j(0)} = \pi/2 - (\alpha_1 + \ldots + \alpha_{j(0)-1})$ scanning the remaining segment of $C_{0,j(0)}$ of length $\alpha'_{j(0)} d_{0,j(0)} \leq \alpha'_{j(0)}(d_{0,1}+\ldots+d_{j(0)-1,j(0)})$. At this point Env switches over to $C_{1,j(0)}$ and thus we continue with turning the ray by the angle $\alpha''_{j(0)} = \alpha_{j(0)} - \alpha'_{j(0)}$ scannnig a segment of length $\leq \alpha''_{j(0)}(d_{1,2}+\ldots+d_{j(0)-1,j(0)})$.

The further procedure should be clear: The rays $\overrightarrow{x_{j(0)+1}, x_{j(0)}}, \ldots \overrightarrow{x_{j(1)-1}, x_{j(1)-2}}$, will be turned by the angles $\alpha_{j(0)+1}, \ldots, \alpha_{j(1)-1}$ whereas the turn of $\overrightarrow{x_{j(1)}, x_{j(1)-1}}$ is again split into two parts with $\alpha'_{j(1)} = \pi/2 - (\alpha_1 + \ldots + \alpha_{j(1)-1}$ and $\alpha''_{j(1)} = \alpha_{j(1)} - \alpha'_{j(1)}$ and so on. Remark that if $j(i) = j(i+1) = \ldots = j(i+k)$ then one has to split the turn of $\overrightarrow{x_{j(i)}, x_{j(i)-1}}$ into $k+1$ parts instead of splitting k different turns into two parts each.

Summing up the estimations for the lengths of the halfcircle segments it turns out that any $d_{i-1,i}$ $(1 \leq i \leq n)$ will be multiplied by $\pi/2$ what proves the first claim.

To prove the second claim let p' be the path $x_0, x_1, \ldots, x_k, x$ and Env' be the part of Env from x_0 to x. W.l.o.g. we may assume that x sees x_n. In the example of Figure 7 we have $n = 3$ and $k = 1$.

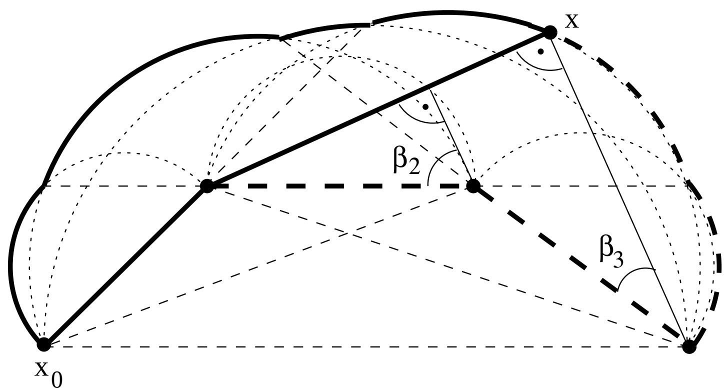


Figure 7: Initial part of an envelope.

As before we scan Env' by a turning ray which is tangent to p and we stop when x is reached. Summing up these estimates we get

$$|Env'| \leq \pi/2 \sum_{i=1}^{k} d_{i-1,i} \; + \sum_{i=k+1}^{n} \beta_i d_{i-1,i}$$

where β_i is the angle between $\overrightarrow{x_i, x_{i-1}}$ and the perpendicular from x_i to the line segment $[x_k, x]$. On the other side the perpendiculars subdivide this line segment and we have $|[x_k, x]| = \sum_{i=k+1}^{n} \sin \beta_i d_{i-1,i}$. Now the second claim follows from the fact that $\beta / \sin \beta \leq \pi/2$ for any $\beta > 0$.

□

Suppose we have to explore a right corner v visible from x_0. The last point on the shortest path from x_0 to the cut $\gamma(v)$ is called the *closest point* on this cut and denoted subsequently by $c_{x_0}(\gamma(v))$.

Clearly, if the halfcircle $C_l(x_0, v)$ is contained in P then $c_{x_0}(\gamma(v))$ is the intersection point of $C_l(x_0, v)$ with $\gamma(v)$ and thus one can reach the closest point walking along this halfcircle. Moreover, the length of the halfcircle segment from x_0 to $c_{x_0}(\gamma(v))$ is at most $\pi/2$ times the lenght of the shortest path.

On the other hand, walking on a halfcircle $C_l(x_0, v)$ which is not entirely contained in P we eventually reach a point z such that:

(A) Walking further on the halfcircle one would loose the visibility contact to v (there is a left reflex vertex v' on the line between z and v); or

(B) It is impossible to continue the walk on $C_l(x_0, v)$ since z is on some polygon edge e, see Figure 8.

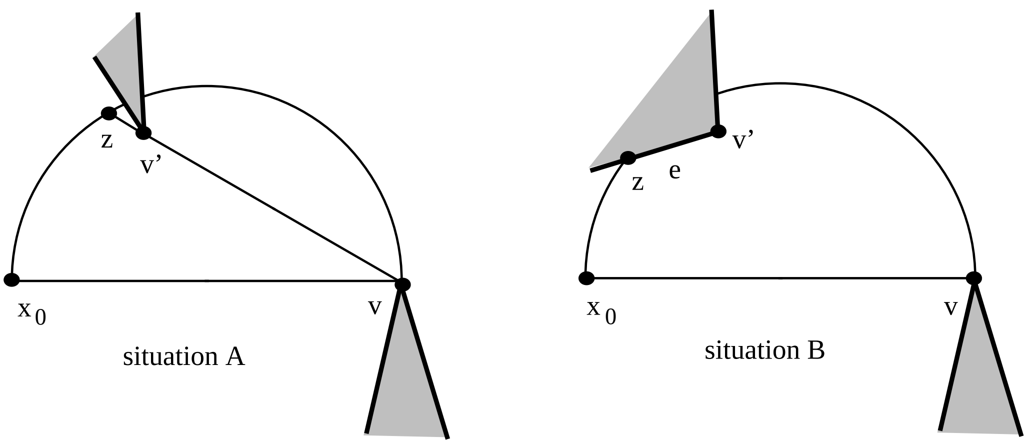


Figure 8: Obstacles on a halfcircle walk.

One can resolve such conflicts by walking directly towards v' in situation **A** or walking along the edge e towards its right vertex v' in case **B**. Then one starts walking on $C_l(v', v)$ as long as the closest point $c_{x_0}(\gamma(v))$ might be on this halfcicle. Note that by Fact 1 this is possible only until $C_l(v', v)$ intersects $C_l(x_0, v)$. At the intersection point one must switch back to $C_l(x_0, v)$. Compare with Figure 9 for illustration.

Based on this observation our halfcircle strategy HCS(x, v) will compute a path from a point x to the closest point $c_x(\gamma(v))$ on the cut of a right corner v which is visible from x.

The basic data structure used is a list L of polygon vertices. In the beginning it consists of x only. In the end, it contains all corners on the shortest path from x to $c_x(\gamma(v))$.

Let in the following y always denote the current last point in L and z the current position on the path generated by the strategy.

Strategy HCS(x, v)

1. Walk on $C_l(y, v)$

   - Situation **A** occurs, then goto 2.
   - Situation **B** occurs, then goto 3.
   - If z is on a halfcircle $C_l(y', v)$ for some ancestor y' of y in L (if this holds for several elements from L let y' be the first of them), then delete all successors of y' in L and goto 1.

2. If z is on the cut $\gamma(v)$, then goto 5.

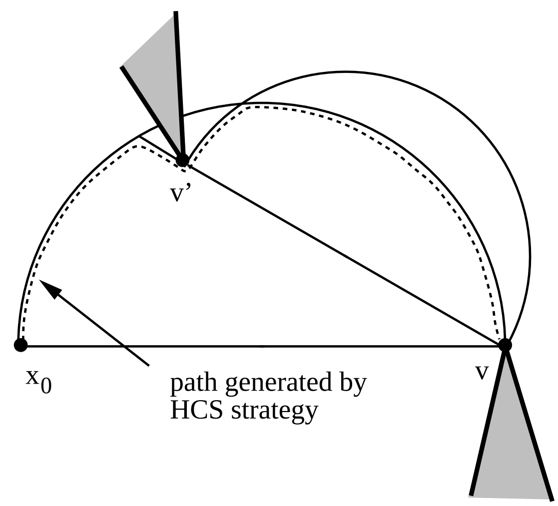


Figure 9: Switching over to a new halfcircle.

3. Let v' be the reflex vertex on the line between z and v. Walk directly towards v'. Goto 4.

4. Walk on the edge e inside $C_l(y, v)$ until

   - Situation **A** occurs, then goto 2.
   - The corner of e is reached, then goto 4.
   - If z is on a halfcircle $C_l(y', v)$ for some y' in L (if this holds for several elements from L let y' be first of them), then delete all successors of y' in L and goto 1.
   - z is on the cut $\gamma(v)$, then goto 5.

5. Search L for the first point visible from z, delete all its successsors from L and add z as the last point to L. Goto 1 (It is possible that one will be confronted with situation **B** immediately).

6. Search L for the first point visible from z, delete all its successsors from L and add z as the last point to L and stop.

Lemma 4.4. The strategy HCS(x, v) is $(\pi/2 + 1)$–competitive and stops at the closest point $c = c_x(\gamma(v))$.
Proof. The second assertion follows from the remarks above.

Next we observe that any occurence of situation **B** can be replaced by type **A** obstacles such that the resulting HCS(x, v)–path is longer. It is therefore sufficient to analyse situation **A** when estimating the HCS(x, v)–path length. This path consists of several halfcircle segments and some straight line segments. By Fact 4.1 the length of any halfcircle segment s is $\alpha_s d_s$ where d_s is the diameter of the halfcircle and α_s is the angle of the wedge from v enclosing the segment s. Then the total length l_1 of all halfcircle segments is bounded by αd where d is the distance from x to v and α is the angle between $\overrightarrow{v,x}$ and $\overrightarrow{v,c}$. Furthermore, the Euclidean distance between x and c is $d \sin \alpha$ and thus $|p(x, c)| \geq d \sin \alpha$. This implies

$$l_1 \leq \alpha d \leq \frac{\alpha}{\sin \alpha} |p(x, c)| \leq \pi/2 |p(x, c)|.$$

We note that any straight line segment on the HCS(x, v)–path is directed towards v and thus the segment length is the difference of the distances from start and end point to v. Since the distance to v decreases along the whole HCS(x, v)–path, we can estimate the total length l_2 of all straight line segments by $d - d'$ where $d = |p(x, v)|$ and $d' = |p(c, v)|$. By the triangle inequality we get

$$l_2 \leq |p(x, v)| - |p(c, v)| \leq |p(x, c)|$$

Together we have $l_1 + l_2 \leq (\pi/2 + 1)|p(c, v)|$.

□

4.2 Scout$_1$–Learning a Right Polygon

Now let us consider a right subpolygon P' of P with a start point $x \in bd(P) \cap P'$. In particular x can stand for any right transit vertex defining the entrance to a room R_x and P' can stand for R_x^1 in this case. Let L_r be the list of all right vertices in P' in counterclockwise order (starting from x on the boundary of P'). Watching from x a robot sees an ordered sublist of L_r only.

Our Scout$_1$–strategy solves the following tasks:

- Generating the complete list L_r;
- Exploring all vertices in L_r;
- Generating a list of all vertices from $L_r \cap T$.

Thus we need three dynamic sublists of L_r: a list L_r^{di} for all discovered vertices, L_r^{un} for discovered but still unexplored vertices, and L_r^T for all transit vertices in L_r.

Outline of the Scout$_1$–strategy:

Scout$_1$ has the following basic structure:

1. Start from the current position and explore the last unexplored vertex v in L_r.

2. Walk along the cut $\gamma(v)$ to the closest point to x (i.e., to the point $c_x(\gamma(v))$).

3. If all right vertices are explored then return to x, else repeat 1.

Since the strategy always aspires to visit the cut of the last unexplored vertex the subpolygon vertices will be explored in clockwise order.

The second and the third stage of the strategy are straightforward. The first stage needs a more detailed explanation. Especially, we have to take into account that the strategy has in general no knowledge about the last unexplored vertex v_{last} in L_r. However, the strategy knows the last still unexplored vertex $v = last(L_r^{un})$ discovered so far.

We can characterize the vertex v_{last} as follows:
Either $v_{last} = v$ or v_{last} is a descendant of v in the discovery tree. Assuming the latter case let v' be the right son of v discovered first. Then we again have that either $v_{last} = v'$ or v_{last} is a descendant of v', and so on.

The basic procedure for exploring $v = last(L_r^{un})$ is the halfcircle strategy HCS. We have, however, to plug in a new feature to manage the strategy's behaviour in situations when the vertex $last(L_r^{un})$ has to be updated, i.e., in the moment when the first right son v' of v is discovered.

Figure 10 illustrates this situation: v' is discovered from point u.

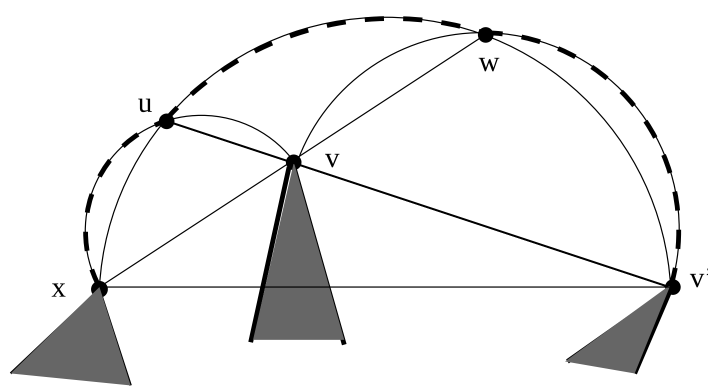


Figure 10: Switching from exploring v to exploring v'.

We have to insert v' into L_r^{un} as the new last vertex and we have, consequently, to switch over from exploring v to the exploration of v'. Again, Fact 4.2 ensures that this can be done without an additional detour since u is on $C_l(x, v')$.

Therefore it is possible to continue the halfcircle strategy replacing v by v' until $C_l(v, v')$ is reached (point w in Figure 10). After this moment the shortest path from x to the cut $\gamma(v')$ has a right turn in v. So it is necessary to keep also a list of all right turns of the shortest path. Reaching $C_l(v, v')$ we insert v into this list and switch over to $C_l(v, v')$.

Clearly, the strategy should be attentive with respect to the follwing three aspects:
a) Whenever a right vertex v'' is discovered such that $f(v'') \in L_r^{un}$ then it is inserted both into L_r and L_r^{un} as the successor of its father. If, moreover, the father is the current last vertex in L_r^{un} then we have to switch over to exploring v'' as discussed above.
b) If some vertex in L_r^{un} gets explored by the way then it is deleted from this list.
c) If a left vertex v'' is discovered such that $f(v'') \in L_r$ then $f(v'')$ is inserted into L_R^T. Moreover, if v'' was the first discovered left son of $f(v'')$, i.e., if $v'' = s(f(v''))$ as defined in 2.2, then we set a pointer from $f(v'')$ to v''.

We remark that vertex x has so far been assumed to be the start position of stage 1. This is true for the first call of that procedure. Later, it starts from a point $\bar{x} = c_x(\gamma(\bar{v}))$ ($\bar{v}$ the previous last unexplored vertex which has been deleted from L_r^{un}) and it has to explore the new $v = last(L_r^{un})$. If v is visible from $\bar{x}$ then we can proceed as above.
Otherwise, the shortest path $p(\bar{x}, v)$ is known which has the form $v_0 = \bar{x}, v_1, \ldots, v_l = v$. Let v_j be the last left turn on $p(\bar{x}, v)$ ($j = 0$ if there are no left turns on this path). In this case the first stage splits into two parts:

i) We walk along the shortest path $p(\bar{x}, v_j)$ to v_j.

ii) From there we use the halfcircle strategy which starts with the exploration of v_{j+1} and switches to the exploration of the next vertices $v_{j+2}, v_{j+3}, \ldots$ as soon as their discovery lines are reached. One has to switch further if v has a right son which was undiscovered so far, and so on. Again, the first stage is finished when we reach the cut of a vertex which has no right sons.

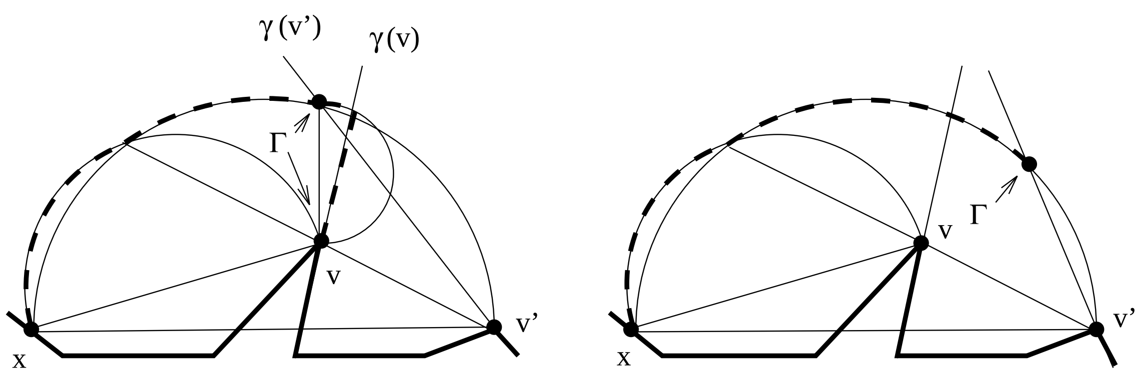


Figure 11: Important cuts and important points.

If a vertex v gets explored by the way then its cut $\gamma(v)$ is called *unimportant*. More precisely $\gamma(v)$ is unimportant if one of the following two situations holds:

1. $v \neq last(L_r^{un})$ at the moment of its exploration,
2. $v = last(L_r^{un})$ at the moment of its exploration and at the same moment a right son v' of v is discovered. This is possible only if v' is the next polygon vertex in counterclockwise order.

Otherwise the cut $\gamma(v)$ will be called an *important cut* and $c_x(\gamma(v)$ an *important point*. This definition is motivated by the fact that the path from an important cut to the next important one is exactly that generated during phase 1 of Scout$_1$. Figure 11 illustrates the concept of important cuts and points. It shows two similar local scenes in a polygon. In Figure 11a there are two important cuts and we observe that after reaching $\gamma(v)$ the strategy moves to v since $v = c_x(\gamma(v))$ in the example.

Let $\Gamma = \{c_x(\gamma(v) \mid \gamma(v)\ \textit{is important}\} \cup \{x\}$ denote the set of all important points. Let $c_1, c_2, \ldots, c_k$ be the elements of Γ in the order induced by the strategy. Remark that these points are not necessarily in relatively convex position as indicated in Figure 12.

We note that the Scout$_1$–path p_i between c_i and c_{i+1} consists of a phase 1 walk and a phase 2 walk which traverses a segment on the cut.

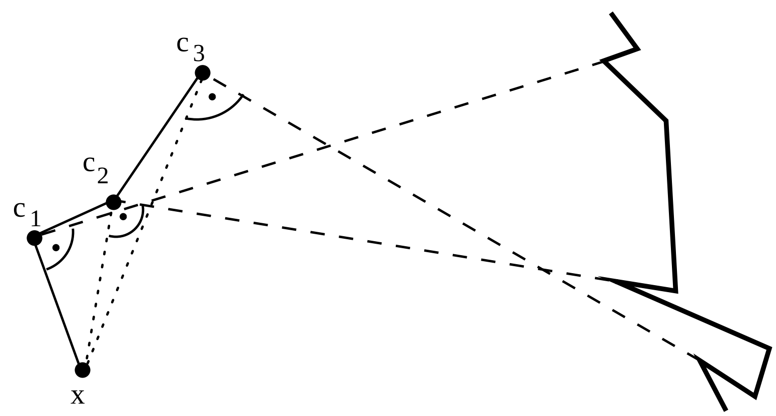


Figure 12: Important points which are not in relatively convex position.

4.3 The Length of a Scout$_1$–Tour

Let $\Gamma = \{c_0 = x, c_1, c_2, \ldots, c_n\}$ be the the set of important points in R_x^1 in the order they are visited by Scout$_1$. $H(\Gamma)$ denotes the relative convex hull of the point set Γ in the polygon P. Let $\Gamma' = \{c_{i_0}, c_{i_2}, \ldots, c_{i_k}\}$ be the set of all important points which are on the boundary of $H(\Gamma)$, where $c_{i_0} = c_0 = x$. The estimation of the length of a Scout$_1$–tour is based on the following chain of estimations formalised below in Propositions 4.5, 4.6, 4.7:

- Length of the Scout$_1$–path between two consecutive important points versus shortest path between the two points,

- Length of the shortest tour through all important closest points versus length of their relatively convex hull boundary,
- Length of the hull boundary versus length of the local optimal watchman tour.

Proposition 4.5. The length of the Scout$_1$–path from c_i to c_{i+1},for $i = 0, \ldots, n$, is bounded by $(\pi/2 + 2)|[c_i, c_{i+1}]|$, where we set $c_{n+1} = x$.

Proposition 4.6. For any $0 \leq j \leq k$ and the corresponding closest point sequence $c_{i_j}, c_{i_j+1}, \ldots, c_{i_{j+1}}$ with $c_{i_{k+1}} = x$ we have $|[c_{i_j}, c_{i_j+1}]| + \ldots + |[c_{i_{j+1}-1}, c_{i_{j+1}}]| \leq \sqrt{2}|[c_{i_j}, c_{i_{j+1}}]|$.

Proposition 4.7. Let W_x^{loc} be the local optimal tour whichs learns R_x^1 in P starting from x. Then there is a relatively convex set $\bar{R}$ in P such that $\Gamma \subset \bar{R}$ and the length of the boundary of $\bar{R}$ is bounded by $(\pi/2 + 1)|W_x^{loc}|$.

Before proving these propositions we remark that together they imply the following bound.

Theorem 4.8. The length of the Scout$_1$–tour which explores R_x^1 is bounded by $\sqrt{2}(\pi/2 + 2)(\pi/2 + 1)|W_x^{loc}|$.

□

Proof of Proposition 4.5. We have to analyse the length of a Scout$_1$–path q between two consecutive important points c_i and c_{i+1}. Starting from c_i let $v' = last(L_r^{un})$ denote the last unexplored vertex out of the set of discovered vertices and let v the last undiscovered vertex of all vertices in R_x^1. Consider the shortest path $p(c_i, v)$ which has the form $v_0 = c_i, v-1, \ldots, v_l = v$. Let v_j be the last left turn on $\pi(c_i, v)$ ($j = 0$ if there are no left turns on this path) and let j' be the index such that $v' = v_{j'}$ then $j < j' \leq l$ and moreover for any j'' with $j \leq j'' < l$ the vertex $v_{j''+1}$ is the first right son of $v_{j''}$. Recall that q consists of the following three parts q_1, q_2, q_3:

- $q_1 = p(v_0, v_j)$.
- q_2 is the "HCS–part" which starts with the exploration of v_{j+1} and switches over to exploring vertices $v_{j+2}, v_{j+3}, \ldots$ as soon as their discover lines are reached. This part ends at $c'_{i+1} = c_{c_i}(\gamma_v)$.
- $q_3 = [c_{c_i}(\gamma_v), c_x(\gamma_v]$ is a line segment on γ_v.

We will prove the following estimations:

$$\begin{array}{ll} (1) & |q_1| \leq |p(v_0, v_j)| \\ (2) & |q_2| \leq (\pi/2 + 1)|p(v_j, c_{c_i}(\gamma(v))| \\ (3) & |q_3| \leq |p(c_i, c_{i+1})| \end{array}$$

Since q_1 is an initial part of $p(v_0, c'_{i+1})$ (1) and (2) together imply

$$(4) \quad |q_1| + |q_2| \leq (\pi/2+1)|p(v_0, c'_{i+1})| \leq (\pi/2+1)|p(c_i, c_{i+1})|$$

Now (3) and (4) together imply the desired bound:

$$(5) \quad |q| \leq (\pi/2+2)|p(c_i, c_{i+1})|$$

The first estimation is straightforward (equality).

Next we prove (3) using a projection argument. Consider the projection of the open segment (c'_{i+1}, c_{i+1}) into the region enclosed by this segment, $p(c_i, c'_{i+1})$, and $p(c_i, c_{i+1})$. If the projection would meet $p(c_i, c'_{i+1})$ this would contradict that c'_{i+1} is the closest point to c_i on the cut.

It remains to prove (2): Note that the shortest path $v_j, v_{j+1}, \ldots, v_l = v$ has only right turns. We start with the easier case that the path generated by the halfcircle strategy is not disturbed by obstacles from the left side. Then q_2 is an initial part of the envelope over the set of halfcircles $C_l(v_{j'}, v_{j''})$ for all $j \leq j' < j'' \leq l$. Thus, applying Fact 4.3 we get even a better estimation: $|q_2| \leq (\pi/2)|p(v_j, c_{c_i}(\gamma(v))|$.

In the case of obstacles we can repeat the arguments presented in the analysis of HCS-paths in the proof of Lemma 4.4. Therefore we have that the total length of all halfcircle segments on p_2 is again bounded by $(\pi/2)|p(v_j, c_{c_i}(\gamma(v))|$ and the total length of all straight line segments used to jump from some halfcircles to vertices of obstacles is bounded by $|p(v_j, c_{c_i}(\gamma(v))|$.

□

The proof of Proposition 4.6. mainly uses the following lemma.

Let $x, y_0, \ldots, y_m$ be points in the plane numbered such that the rays $\overrightarrow{x, y_i}$ are clockwise ordered and such that they satisfy the following condition (P):
The points $Y = \{y_1, \ldots, y_{m-1}\}$ lie in the triangle spannend by x, y_1, and y_m and for each $1 \leq i \leq m-1$ we have $y_i \notin int(C_r(x, y_{i-1}))$. Then the following holds.

Lemma 4.9. For $L(Y) = \sum_{i=0}^{m-1} |[y_i, y_{i+1}]|$ we have $L(Y) \leq \sqrt{2}|(y_0, y_m)|$.
Proof. We transform by local operations the set $Y = \{y_1, \ldots, y_{m-1}\}$ into a new set Y' still satisfying condition (P) such that $L(Y) \leq L(Y')$. Eventually we will reach a configuration which trivially implies the claimed inequality. We process the points in Y in decreasing order starting with y_{m-1}. Assume the next point to be processed is y_i:

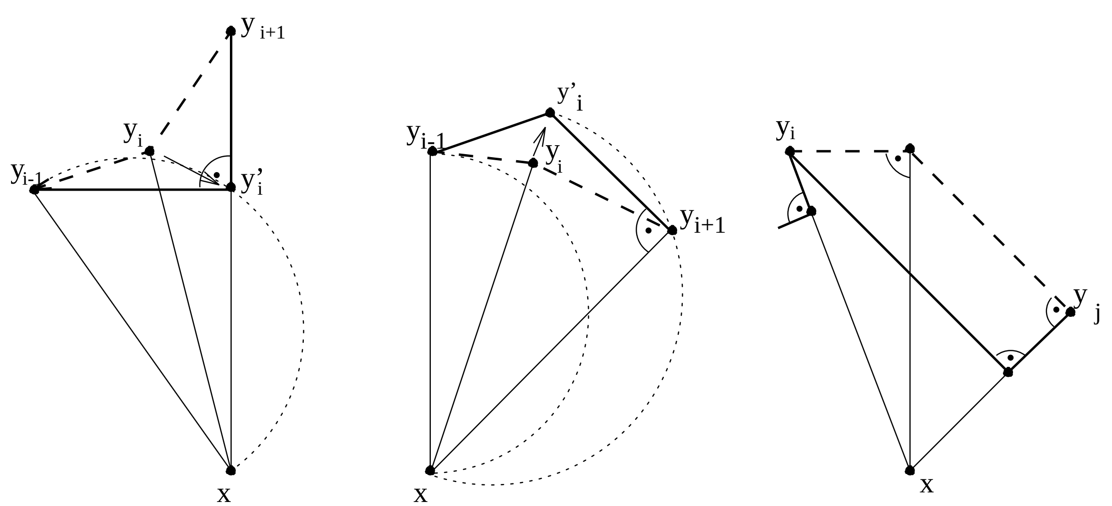
Left Turn: If $y_i \in Y$ defines a left turn, i.e., $[y_i, y_{i+1}]$ is left of $[y_{i-1}, y_i]$ then we replace y_i by the point $C_r(x, y_{i-1}) \cap (x, y_{i+1})$.
This operation preserves obviously property (P) and gives a longer tour. In the end this left turn is a right angle, see Figure 13a.

Right Turn: If y_i defines a right turn we shift it along the ray towards the line (y_0, y_m) until the new position y'_i is such that either $y_{i+1} \in C_r(x, y'_i)$ or $y'_i \in (y_0, y_m)$, compare with Figure 13b. In the latter case the problem splits and we are done while in the first case we proceed with the next point.

Eventually we reach a configuration such that:
If y_i is a left turn then $y_i \in [x, y_{i+1}]$, otherwise $[y_i, y_{i+1}]$ and $[x, y_{i+1}]$ define a right angle.

The last step is to substitute a maximal chain of at least two consecutive right turns $y_i, \ldots, y_j$ by the perpendicular from y_i to (x, y_{j+1}), see Figure 13c. Now we are in a situation where left and right turns alternate and the inequality follows.

□



(a) Transforming a Left Turn (b) Transforming a Right Turn . (c) Consecutive Right Turns.

Figure 13: Estimating the Important Point Tour.

Proof of Proposition 4.6. The claim now follows from the previous lemma with $Y = \{c_{i_j+1}, \ldots, c_{i_{j+1}}\}$. The property (P) is fulfilled for this point set what follows directly from the fact that all points in Y are important.

□

Proof of Proposition 4.7. Let us assume that the optimal local tour $W = W_x^{loc}$ is given with clockwise orientation.
Note that Γ is a set of closest points on some cuts which are visited by W. One could try to define $\bar{R}$ to be the relative convex hull of $H(W) \cup \Gamma$. However, it seems difficult to estimate the boundary length of this set. Instead we extend $H(W)$ to a larger set Ext defined by

$$Ext = \{c \mid \text{ there is a line segment } l \text{ in } P \text{ such that } c = c_x(l) \notin [W] \text{ and } l \cap W \neq \emptyset\}.$$

For any $c = c_x(l) \in Ext$ let $w = w(l)$ be the first point on W reached following l from c towards $H(W)$. Further, let $\overline{w} = \overline{w}(l)$ be the last point on the shortest path $p(x,c) \cap W$.

Now we are going to derive necessary conditions (conditions 1, 2, 3 below) for a point to be element of Ext.

Let us first subdivide W at all points where it turns to the left (these are necessarily reflex vertices) and, moreover, at all points on W which are local maxima or minima of the function $|p(x,-)|$. Note that such a local maximum is always a right turn and a local minimum is either a left turn or closest point to x on a line segment of W. Thus we obtain a partition of W into *monotone pieces*, i.e., the distance from x to a point moving along such a piece is a monotone function.

By an elementary case inspection we get

Condition 1: If $c = c_x(l) \in Ext$ then $w(l)$ and $\overline{w}(l)$ belong to the same monotone piece of W.

Consequently, it suffices to consider separately those parts of Ext corresponding to monotone pieces of W.
From the definitions we have that $|p(x,\overline{w})| \leq |p(x,c)| \leq |p(x,w)|$ and $|p(x,\overline{w})| + |p(\overline{w},c)| = |p(x,c)|$.
This implies $|p(\overline{w},c)| \leq |p(x,w)| - |p(x,\overline{w})|$.
Applying both the triangle inequality and the fact that the distance $dist_W$ on W is not smaller than the distance in P we get

Condition 2: $|(\overline{w},c)| \leq |(\overline{w},w)| \leq dist_W(\overline{w},w)$.

Finally, let $e_1, e_2, \ldots, e_k$ be a monotone increasing piece of W. Then there are only right turns on this piece. Let α_i denote the angle between e_{i+1} and the prolongation of e_i. Assume that for some $c = c_x(l)$ we have $\overline{w} = \overline{w}(l) \in e_j$ and $w = w(l) \in e_m$. Let α be the angle between e_i and $p(\overline{w},c)$. We are going to prove

Condition 3: $\alpha + \sum_{i=j}^{m-1} \alpha_i \leq \pi/2$.

Consider the region encircled by $p(\overline{w},c)$, the part of W from $\overline{w}$ to w and the line segment $[c,w]$. It is an $(m-j+3+j)$–gon where j is the number of turns on $p(\overline{w},c)$. Since $p(\overline{w},c)$ is the shortest path from $\overline{w}$ to l it is straightforward that all turns on this path are left turns (with inner angles $\pi/2+\beta_1,\ldots,\pi/2+\beta_j$) and that the angle β between $p(\overline{w},c)$ and $[c,w]$ is at least $\pi/2$. Condition 3 then follows from the formula for the interior angle sum in an n–gon.

Now we can construct a set $\overline{Ext}$ containing Ext:
Let $e_1, e_2, \ldots, e_k$ be a monotone increasing piece of W and let $y_0, y_1, \ldots, y_k$ be the corresponding points such that e_i is the line segment between $[y_{i-1}, y_i]$.

In the rather trivial case $k = 1$ we define $\overline{Ext}$ to be the cycle sector centered at y_0 with radius $|e_1|$ and angle $\pi/2$ from y_1. Obviously, any point c satisfying Conditions 1–3 and such that $w, \overline{w} \in e_1$ is in this set. Moreover, after this extension the new boundary consists of a quarter of a circle and one radius. Thus, its length

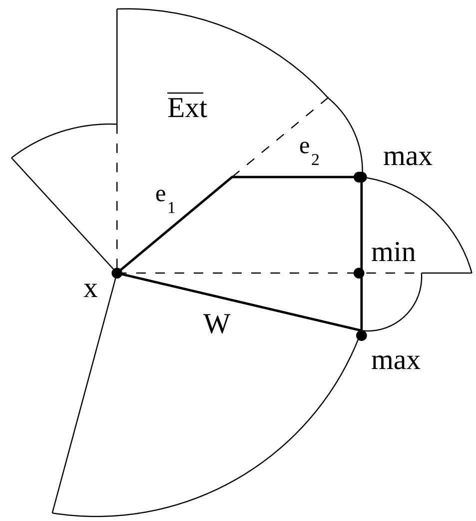


Figure 14: Constructing the region $\overline{Ext}$.

is $(\pi/2+1)r_1$.

In the general case the construction of $\overline{Ext}$ is similar, but a bit more complicated. We note that all α_i are at most $\pi/2$ because otherwise there was a local maximum in y_i.

Start to draw from y_k an arc centered at y_{k-1} with radius $|e_k|$ and angle α_{k-1}. Continue with an arc centered at y_{k-2} with radius $|e_{k-1}|+|e_k|$ and an angle of size $min\{\alpha_{k-2}, \pi/2-\alpha_{k-1}\}$. In the case that this minimum is determined by the first term, i.e., if $\alpha_{k-2} \leq \pi/2-\alpha_{k-1}$, continue with an arc centered at y_{k-3} and radius $|e_{k-2}|+|e_{k-1}|+|e_k|$, and so on.

Otherwise, if the total angle of all arcs drawn so far equals $\pi/2$, then draw a "back line segment" of length $|e_k|$ directed towards the center, and continue with a circle arc having a radius decreased by $|e_k|$. Clearly, one has to reduce the radius by $|e_{k-1}|$ if the total angle of all arcs except the first one is $\pi/2$, and so on.

Figure 14 illustrates this construction. Conditions 1, 2, 3 imply that $Ext \subseteq \overline{Ext}$. Since each radius $|e_i|$ contributes to arcs with a total angle of $\pi/2$ and to one back line segment the length of the new boundary is $(\pi/2+1)(|e_1|+\ldots+|e_k|)$.

Finally one has to study what happens if $\overline{Ext}$ is not fully contained in P. The main observation is that obstacles have the same effect as back line segments. Instead of giving a detailed description of the construction we refer to Figure 15. Again, it is not hard to analyze that the so constructed set $\overline{Ext}'$ contains Ext and has a boundary of length $\leq (\pi/2+1)|W|$.

$\square$

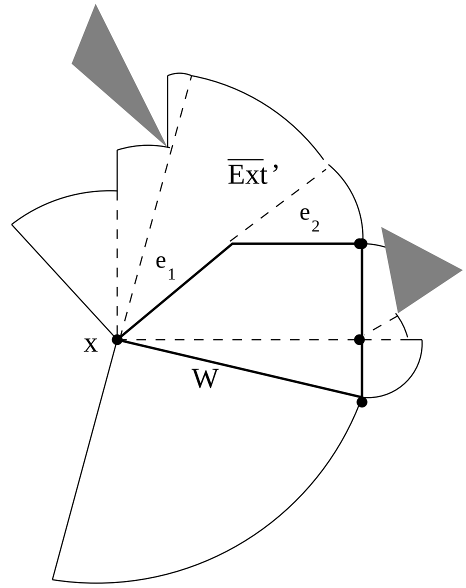


Figure 15: Constructing the region $\overline{Ext}'$ in the presence of obstacles.

4.4 Scout$_2$ – Exploration of a Room

We assume that the right subpolygon R_x^1 of a room R_x with right entrance x has been explored by Scout$_1$. Now we have to explore all connected components of $R_x^2 = R_x \setminus R_x^1$. These are left polygons. We note that at least one left vertex for each component has been discovered by Scout$_1$. These vertices are given by the pointers from L_r^T. Thus, inserting them into a list L_l^{di} in reverse order one can start a left variant of Scout$_1$ with the following modification:
Whenever a right son v' of a left vertex $v \in L_l^{un}$ is discovered then insert v into L_l^T, delete it from L_l^{un}, and set a pointer from v to v'. In other terms this means that v has been recognized to be an exit door of R_x. So in the end L_l^T is the list of all exit doors of R_x. It is straightforward that the components of R_x^2 will be explored in counterclockwise order. One can repeat exactly the same arguments as above to analyze the length of the Scout$_2$–tour.

5 Conclusions

1. We have proved an upper bound of $10\sqrt{2}(\pi/2+1)(\pi/2+2)+3 \leq 133$ on the competitive ratio of the Scout–and–March strategy SAM.
It should be possible to improve some of the factors noticably. So we address as a first problem to prove better upper bounds on the competitive ratio for learning simple polygons by SAM or alternative on–line strategies.

2. It is not hard to adapt SAM to situations where the starting point x_0 is not on

the polygon boundary. The only difference appears during the left and the right exploration Scout_1 and Scout_2 of the first room R_{x_0}. The problem is that an on–line strategy does not know how to break optimally the cyclic order of the right (left) vertices visible from x_0 into a linear order. The detour caused by this fact yields a larger competitive ratio. An easy analysis shows that it is sufficient to add $2(\pi/2+1)^2$ as a summand to the constant above.

3. Many of the definitions and techniques introduced in this paper rely on the assumption that P is a simple polygon. A typical example is the basic notion of left and right vertices as well as the halfcircle strategy. So it needs some more effort to prove the claim in [9] stating that there is a competitive strategy for learning polygons with a bounded number of holes.

4. The best known lower bound on the competitive ratio of on–line learning simple polygons is $\sqrt{2}$ for watchman paths and $\frac{\sqrt{2}+1}{2}$ for watchman tours. In the case of polygons with one hole one can easily improve these lower bounds to 3 for watchman paths and 2 for watchman tours.
It is known that there is no competitive on–line strategy for learning polygons with an arbitrary number of arbitrary holes.

5. The main and most interesting open problem in this field:
Is there a competitive on–line strategy for learning rectilinear polygons with an arbitrary number of rectilinear holes?
If yes, then one could hope for a generalization to polygons with an arbitrary number of “non-flat” holes. Here the notion “non-flat” could for example be formalized as a lower bound on the interior angles of the holes.

References

[1] H. Abelson and A. diSessa. *Turtle Geometry*. MIT Press, Cambridge, MA, 1980.

[2] G. Asser. Bemerkungen zum Labyrinth-Problem. *EIK*, 13(4/5):203–216, 1977.

[3] R. Baeza-Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and Computation*, 106:234–252, 1993.

[4] A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. In *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, pages 494–503, 1991.

[5] S. Carlsson and H. Jonsson. Computing a shortest watchman path in a simple polygon in polynomial-time. In *Proc. 4th Workshop Algorithms Data Struct.*, Lecture Notes in Computer Science. Springer-Verlag, 1995.

[6] S. Carlsson, H. Jonsson, and B. J. Nilsson. Finding the shortest watchman route in a simple polygon. In *Proc. 4th Annu. Internat. Sympos. Algorithms*

Comput. (ISAAC 93), volume 762 of *Lecture Notes in Computer Science*, pages 58–67. Springer-Verlag, 1993.

[7] W.-P. Chin and S. Ntafos. Watchman routes in simple polygons. *Discrete Comput. Geom.*, 6(1):9–31, 1991.

[8] A. Datta and C. Icking. Competitive searching in a generalized street. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 175–182, 1994.

[9] X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 298–303, 1991.

[10] X. Deng, T. Kameda, and C. H. Papadimitriou. How to learn an unknown environment I: the rectilinear case. Technical Report CS-93-04, Department of Computer Science, York University, Canada, 1993.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, NY, 1979.

[12] K. Ghosh and S. Saluja. Optimal on-line algorithms for walking with minimum number of turns in unknown streets. Technical Report TCS-94-2, Tata Institute of Fundamental Research, Bombay, 1994.

[13] F. Hoffmann. On the rectilinear art gallery problem. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, volume 443 of *Lecture Notes in Computer Science*, pages 717–728. Springer-Verlag, 1990.

[14] F. Hoffmann, M. Kaufmann, and K. Kriegel. The art gallery theorem for polygons with holes. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 39–48, 1991.

[15] C. Icking and R. Klein. Competitive strategies for autonomous systems. In H. B. et al., editor, *Environment Modelling and Motion Planning for Autonomous Robots.* World Scientific, 1995.

[16] C. Icking and R. Klein. Searching for the kernel of a polygon: A competitive strategy. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 258–266, 1995.

[17] C. Icking, R. Klein, and L. Ma. How to look around a corner. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 443–448, Waterloo, Canada, 1993.

[18] R. Klein. Walking an unknown street with bounded detour. *Comput. Geom. Theory Appl.*, 1:325–351, 1992.

[19] J. M. Kleinberg. On-line search in a simple polygon. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 8–15, 1994.

[20] A. Lopez-Ortiz and S. Schuierer. Going home through an unknown street. In *Proc. 4th Workshop Algorithms Data Struct.*, Lecture Notes in Computer Science. Springer-Verlag, 1995.

[21] V. J. Lumelsky and A. A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 1987.

[22] T. Ottmann, S. Schuierer, and C. A. Hipke. Kompetitive Analyse für Online-Algorithmen: Eine kommentierte Bibliographie. Technical Report 61, Institut für Informatik, Universität Freiburg, 1994.

[23] N. S. V. Rao, S. Kareti, W. Shi, and S. S. Iyengar. Robot navigation in unknown terrains: introductory survey of non-heuristic algorithms. Technical Report ORNL/TM-12410, Oak Ridge National Laboratory, 1993.

[24] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.

[25] I. Sutherland. A method for solving arbitrary wall mazes by computer. *IEEE Transactions on Computers*, C-18(12):1092–1097, 1969.

[26] X. Tan and T. Hirata. Constructing shortest watchman routes by divide-and-conquer. In *Proc. 4th Annu. Internat. Sympos. Algorithms Comput. (ISAAC 93)*, volume 762 of *Lecture Notes in Computer Science*, pages 68–77. Springer-Verlag, 1993.