REACHING THE KERNEL OF AN UNKNOWN POLYGON FAST

L. PALIOS

33-99

Preprint no. 33-99/1999

Department of Computer Science
University of Ioannina
451 10 Ioannina, Greece

Reaching the Kernel of an Unknown Polygon Fast

Leonidas Palios
Department of Computer Science
University of Ioannina, Ioannina, Greece
palios@cs.uoi.gr

Abstract: We consider the following motion planning problem for a point robot inside a simple polygon P: starting from an arbitrary point s of P, the robot aims at reaching the closest point t of P from where the entire polygon P can be seen; the robot does not have complete knowledge of P but is equipped with a 360-degree vision system that helps it "see" its surrounding space. We are interested in a competitive path planning algorithm, i.e., one that produces a path whose length should be no more than a constant c times the length of the shortest off-line path (in this case, $c \times \text{distance}(s,t)$); the constant c is called the competitive factor. In this paper, we present a new strategy that achieves a competitive factor of $\sim$3.126, improving over a 5.48-competitive strategy of Icking and Klein and a 3.829-competitive strategy of Lee et al. Additionally, our strategy possesses two advantages compared to past solutions: first, the first point reached from where the entire polygon P is seen is precisely the closest such point to the starting position, and second, all the points of the path are directly determined in terms of the starting position and of polygon vertices, which implies that an actual robot following the strategy is not expected to deviate much from its course due to numerical error. The competitiveness analysis is based on properties of the class of curves with increasing chords.

Keywords: Motion planning, competitive algorithm, kernel, simple polygon, curve with increasing chords.

1. Introduction.

The field of robot motion planning has received considerable attention during the 1980s, but research intensified in the late 1980s when technological advances paved the way towards adequate autonomous function of robots. This fact, along with the need for autonomous robots to undertake tasks that may be dangerous for humans (areas polluted by chemicals, space exploration, etc.), led to a number of results pertaining to motion planning algorithms in partially known or unknown environments in the 1990s. (A comprehensive survey on motion planning results until the early 1990s can be found in [8].)

The general motion planning problem involves devising strategies which can help a robot to get to a destination point in an environment which is being "discovered" by means of a 360-degrees vision system (and in some early work, by means of tactile sensing). Most motion planning problems are being modeled as two-dimensional problems where the robot is a point moving inside or around polygonal shapes. For example, a robot moving inside a complex of rooms can be modeled as a point moving in the interior of a polygon containing

smaller polygons; this polygon is the geometric image of the floorplan of the complex. It is important to note that modeling a robot as a point (despite the fact that every robot has nonzero mass) is not really restrictive, as real-world problems can be reduced to this formulation by means of transformations of the geometric boundaries of the objects in the robot's world (Minkowski sum, etc; see [4]).

Of course, the application of an exhaustive search strategy of the environment is bound to solve any motion planning problem considered. However, this is not what one wants to have in practical applications. Instead, one would like to have strategies which guarantee that the path traveled by the robot up to its destination is no more than a constant times the length of the shortest path if the environment was completely known. Such strategies are called *competitive* [20], and the ratio of the length of the actual path traveled over the length of the shortest path is called the *competitive factor*. In other words, the competitive strategies guarantee that the effort expended is not far from the optimal. Research results indicated that finding competitive strategies for different motion planning problems exhibits varying degrees of difficulty. In particular, some problems admit competitive solutions (e.g., axis aligned square obstacles [16]), in other problems the competitive factor is bounded by a function of the description size of the environment (but not a constant) (e.g., the "room" problem, the "wall" problem [1]), while finding competitive solutions for yet other problems seems a very difficult task (the Canadian traveler problem is PSPACE complete [16]).

Early work on motion planning addressed problems where the location of the destination was known, although the locations and shapes of the obstacles in between were not. More recent work dealt with problems where the robot does not know the exact location of the destination but it can recognize it as soon as it sees it. An interesting problem in the latter case, the problem of locating the target t in an *(s-t)-street* (a special type of polygon), has been the focus of considerable research effort [12, 10, 19], which yielded competitive strategies with factors less than 6, culminating to an optimal strategy with competitive factor equal to $\sqrt{2}$ [10, 19]. Competitive strategies have also been proposed for target location in "generalized" streets, (namely, the G-streets [3, 14], the HV-streets [2], and the θ-streets [2]), none optimal so far to the best of our knowledge.

In this paper, we consider the problem of planning the path of a robot inside a polygon from any given starting position to a point from where the entire polygon can be seen; in fact, the closest such point to the starting position is sought. This is the problem of reaching the *kernel* of a polygon, and is what a mechanical guard is called to solve in order to position itself so that it watches a certain territory. The problem has been considered by Icking and Klein [9] who described a strategy to reach the closest point of the kernel achieving a competitive factor of $\sim$5.48. They also showed that no competitive factor less than $\sqrt{2}$ can be achieved. A different strategy with a competitive factor of $\sim$3.829 was later described by Lee et al. [13], while López-Ortiz and Schuierer [15] improved the lower bound to $\sim$1.48. López-Ortiz and Schuierer also noted that the competitive factor of [9] is not guaranteed for negative instances (i.e., when the polygon has empty kernel, in which case the competitive factor is defined as the ratio of the path length up to the point where the robot realizes that the kernel is empty over the length of the shortest off-line path to come to the same conclusion) and described a strategy that is guaranteed to work even in this case at a competitive factor of $\sim$46.35.

Our work contributes a new strategy for reaching the kernel of an unknown polygon with nonempty kernel which achieves a competitive factor of $\sim$3.126. The path consists of line segments and circular arcs whose total number is linear in the size of the polygon P.

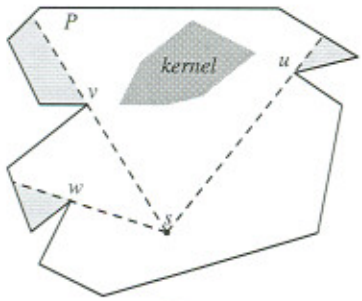


Figure 1

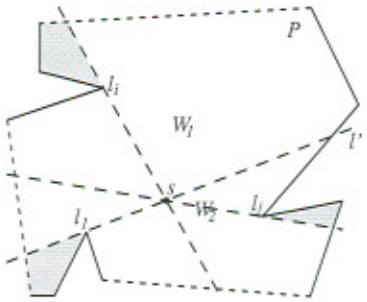


Figure 2

Our strategy is designed so that the robot walks into the kernel at precisely the point that is closest to the starting position; additionally, it has the advantage that any point of the course is determined by the starting position of the robot and vertices of P, and therefore an actual robot following the strategy is not expected to deviate much from its course due to accumulated numerical errors. The competitiveness analysis is based on properties of the class of curves with increasing chords [18]. Experimental results suggest that the strategy performs better than the theoretical competitive factor.

The paper is structured as follows. In Section 2 we review the terminology that we use throughout the paper, and in Section 3 we outline our strategy. Sections 4 and 5 establish the competitive factor of our strategy and some of the properties of the resulting path. In Section 6 we conclude with final remarks and open questions.

2. Terminology.

A *simple polygon* is the region enclosed by a single closed polygonal line that does not intersect itself; thus, a simple polygon does not have "holes" in it. A simple polygon P is *star-shaped* if there is a point p of P such that the line segment that connects p with any other point of P lies entirely in P. The set of all such points p is called the *kernel* of the polygon [4, 17]. If we define the *inner halfplane* of an edge as the closed halfplane which is defined by the edge and contains all the points of P in a sufficiently small neighborhood of the edge's midpoint, then the kernel of P is equal to the intersection of the inner halfplanes of all the edges of P and is therefore convex.

We will follow the terminology of Icking and Klein [9]; we briefly summarize it in this paragraph. From its starting position s, the robot probably does not see parts of the polygon P in which it stands; if the robot sees all of P, then s belongs to the kernel and the robot need not move. The hidden portions of the polygon are called *caves*. Each cave is adjacent to a reflex vertex of P, whose very existence creates the cave; these reflex vertices are called *constraint vertices* (Figure 1). A cave (associated with a constraint vertex v) is characterized as either *left* if it lies to the left of the directed line $\vec{sv}$, or *right* otherwise. By extension, we say that a vertex is a *left constraint vertex* if it is a constraint vertex associated with a left cave, and similarly for a *right constraint vertex*. In Figure 1, the vertices v and w are left constraint vertices, and the shaded regions next to them are the associated caves; the vertex u is a right constraint vertex. For each of the constraint vertices v, we define its *inner halfplane* with respect to the current position p as the closed halfplane which is delimited by the line $\overline{pv}$ and does not contain the corresponding cave. Clearly, any point that sees the cave next to the constraint vertex v has to belong to the inner halfplane of v.

From its starting position, the robot may detect zero or more left caves and zero or more right caves. If the robot sees at least one left cave, the following lemma holds.

Lemma 2.1. *Suppose that from its starting position s in a simple polygon P the robot detects one or more left caves next to the constraint vertices $l_1, \ldots, l_k$ ($k \geq 1$). Suppose further that no left constraint vertex exists such that the closure of the complement of its inner halfplane contains all the left constraint vertices. Then, the kernel of P is empty.*

Proof: For $k = 1, 2$, there is always a left constraint vertex l such that the closure of the complement of its inner halfplane contains all the left constraint vertices: this is trivially true if $k = 1$; if $k = 2$, l is the left constraint vertex that belongs to the inner halfplane of the other constraint vertex. So, it must be the case that $k \geq 3$. Let us consider l_1; then, in accordance with the statement of the lemma, there exist left constraint vertices which belong to the interior of l_1's inner halfplane; let l_i be that among these vertices such that the clockwise angle $\widehat{l_1 s l_i}$ is maximum (Figure 2). Then, there are no left constraint vertices in the interior of the wedge W_1 which is the intersection of l_1's and l_i's inner halfplanes. On the other hand, there exists at least one left constraint vertex l_j belonging to the interior of l_i's inner halfplane; since W_1 is empty of left constraint vertices, l_j belongs to the interior of the wedge W_2 which is the intersection of the inner halfplanes of l_i and l_j, or on the halfline sl'. In the former case, the intersection of the inner halfplanes of l_1, l_i and l_j is $\{s\}$; since the kernel K is a subset of the inner halfplanes of all the left constraint vertices, then $K \subseteq \{s\}$, which implies that $K = \emptyset$, because s does not see the entire polygon. In the latter case, the intersection of the inner halfplanes of l_1, l_i and l_j is the halfline sl'; the kernel is empty again, since no point of the halfline can see l_1's associated cave. ∎

A similar lemma holds for the right constraint vertices. Therefore, if the conditions of Lemma 2.1 hold, we need do nothing, since the polygon has empty kernel. Otherwise, there is a left constraint vertex such that the closure of the complement of its inner halfplane contains all the left constraint vertices and it is unique (if there are more than one vertices collinear with s then we choose the one farthest away from s); we call this vertex *maximal left constraint vertex*. In Figure 1, v is the maximal left constraint vertex. In a similar fashion, we have the *maximal right constraint vertex*.

We include below another lemma, which establishes a fact about the spatial relationship of the left and right constraint vertices in a polygon with nonempty kernel.

Lemma 2.2. *Suppose that from its starting position s in a simple polygon P the robot detects one or more left caves next to the constraint vertices $l_1, \ldots, l_k$ ($k \geq 1$) and one or more right caves next to the constraint vertices $r_1, \ldots, r_m$ ($m \geq 1$). Let l_1 be the maximal left constraint vertex and let $\overline{H(l_i)}$ denote the complement of the inner halfplane of l_i. Then, if there exists a right constraint vertex belonging to $\overline{H(l_1)} - \cap_i \overline{H(l_i)}$, the kernel of P is empty.*

Proof: Let l_j be the left constraint vertex for which the counterclockwise angle $\widehat{l_1 s l_j}$ is maximum. Then, in accordance with the definition of the maximal left constraint vertex, all the left constraint vertices belong to the wedge which is the intersection of l_j's inner halfplane and the complement of l_1's inner halfplane. In fact, this wedge is precisely the intersection $\overline{H(l_1)} - \cap_i \overline{H(l_i)}$; the wedge contains the halfline sl_j, but not the halfline sl_1. If there exists a right constraint vertex r in $\overline{H(l_1)} - \cap_i \overline{H(l_i)}$, then no matter whether it lies in the interior of this wedge (Figure 3), or on the halfline sl_j, the intersection of the inner halfplanes of l_1, l_j and r is precisely $\{s\}$, and the kernel is empty. ∎

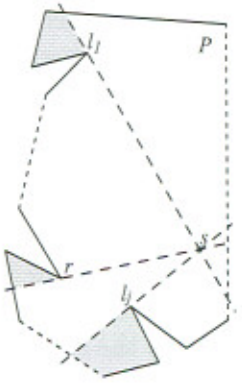


Figure 3

A similar lemma holds for the right constraint vertices. The lemma implies that in a polygon with nonempty kernel, the left and right constraint vertices are not "intermixed" and this is why in papers on this problem which assume polygons with nonempty kernel, figures show the left and the right constraint vertices all gathered on the left and on the right of the polygon boundary respectively.

Crucial in the analysis of our strategy is the notion of a *curve with increasing chords*; a curve has increasing chords if $|ad| \geq |bc|$ for any four points a, b, c, d lying on the curve in that order ($|pq|$ denotes the length of the line segment connecting p and q). For a plane curve with increasing chords, Rote proved that

Lemma 2.3. [18] *The length of a plane curve with increasing chords connecting two points a and b does not exceed $\frac{2\pi}{3}$ times the length of the line segment connecting a and b.*

We will also refer to the quadrants associated with a point p. If p is a point different from the starting position s of the robot, then we define the *quadrants* A_p, B_p, C_p and D_p at p as the four closed quadrants determined by the line $\overline{sp}$ (through s and p) and its perpendicular at p: the quadrant A_p is the quadrant that contains s and lies to the right of the directed line $\vec{sp}$, while the other quadrants B_p, C_p and D_p follow quadrant A_p in counterclockwise order around p (Figure 4).

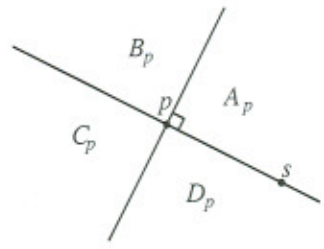


Figure 4

We close this section with a well known geometric fact and another lemma, which will be useful later.

Fact 2.1. *Consider a circle with diameter ab. Then, for the angle $\widehat{apb}$ of the triangle with vertices a, b, and p:*

(i) $\widehat{apb} < \pi/2$ *if p lies outside the circle;*

(ii) $\widehat{apb} = \pi/2$ *if p lies on the boundary of the circle;*

(iii) $\widehat{apb} > \pi/2$ *if p lies inside the circle.*

Lemma 2.4. *Let C_1 be a connected non-self-intersecting curve which does not intersect the line segment connecting its endpoints a and b, and C_2 a convex polygonal line with the same endpoints which lies in the region enclosed by C_1 and the line segment ab. Then, the length of C_2 does not exceed the length of C_1.*

Proof: (Proofs of the above proposition for a convex polygonal line C_1 can be extended to prove this case as well.) Let $a = v_1, v_2, \ldots, v_k = b$ be the vertices of C_2 in order from a to b. We extend each line segment $v_{i-1}v_i$ past v_i until it hits C_1; let u_i be the point of intersection. Because C_2 is convex, no pair of these extensions intersect. Moreover, for each segment $v_{i-1}v_i$, we have that $|v_{i-1}u_i| = |v_{i-1}v_i| + |v_iu_i| \leq |v_{i-1}u_{i-1}| + |\widetilde{u_{i-1}u_i}|$, where $|pq|$ and $|\widetilde{pq}|$ denote the length of the line segment and the length of the curve C_1 between p and q respectively. The lemma follows from the summation of all these inequalities; for $1 < i < k$ the terms $|v_iu_i|$ cancel out, whereas $|v_1u_1| = |aa| = 0$ and $|v_ku_k| = |bb| = 0$. ∎

Angle Notation: Since three points define two angles (which sum up to 2π), in the following, the notation $\widehat{abc}$ (where a, b, c are three non-collinear points) is meant to indicate the smallest of the two corresponding angles.

3. The Strategy.

The basic motivation behind our strategy stems from the study of the simplest case, i.e., a single reflex vertex v whose incident edges are not both visible from the starting position s. Since the robot does not know the direction of the invisible edge e incident upon v, it does not know where the closest point t of the kernel might be. However, no matter what the direction of e is, t belongs to the semicircle with diameter sv, assuming that the semicircle lies in the polygon P (Figure 5). So, a good strategy is to follow this semicircle.

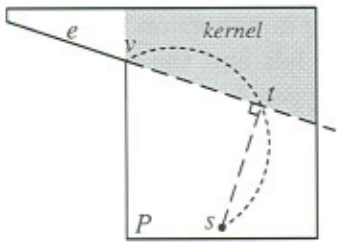


Figure 5

Our strategy is based on this idea. Therefore, the path of the robot consists of circular arcs and line segments; each circular arc belongs to a circle with diameter sp, where s is the starting position and p is a constraint vertex. This strategy makes the robot reach the kernel at its closest point to s.[1]

We first consider the one-sided case, where there are only left or only right caves; our strategy for the general case consists of applying the one-sided case strategy twice, first for the left caves until we see them all, and then for the right caves (if needed).

3.1. The one-sided case. Without loss of generality, we consider the case where there are only left caves (the case where we have only right caves is similar). Until the robot sees all the left caves, there exist left constraint vertices and among them a maximal left constraint vertex, which may change as the robot moves; at any given time, the robot's objective is to try to see the cave incident upon the currently maximal constraint vertex.

Initially, the robot finds the maximal left constraint vertex v_0 as seen from the starting position s and starts following the semicircle with diameter sv_0. The two fundamental cases that characterize the robot's path are:

1. *A new maximal constraint vertex u is discovered.* Then, the robot will start following the semicircle with diameter su (Figure 6: point a). Interestingly, the current location of the robot belongs to both semicircles.
2. *The cave next to the currently maximal constraint vertex u becomes visible.* This implies that the second edge e incident upon u has become visible as well. Then, the robot at its current position, say, b, finds the new maximal constraint vertex. If no such vertex exists, then the entire polygon is visible and the robot has achieved its goal. If such a vertex exists —let it be v— and v is a constraint vertex just seen for the first time (for example, if v is the other endpoint of e), then we execute the previous case. The remaining possibility is if v is a constraint vertex that has already been seen, in which case the robot walks along the line segment bu trying to reach (if possible) the semicircle with diameter sv (Figure 6: points b and c).

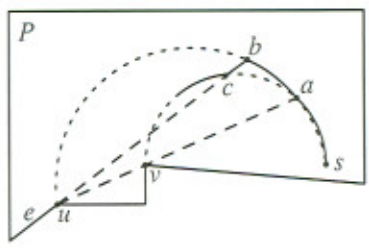


Figure 6

[1] It must be noted that this strategy is not optimal for the simple case of a single reflex vertex; it yields a worst-case competitive factor of $\pi/2 \simeq 1.57$. See [11], for a proof that the optimal competitive factor is $\sim$1.212, and for a strategy achieving it.

Note that it may be the case that the robot has to reach the currently maximal constraint vertex u in order to see the cave next to u. (This can only happen if u is the maximal left constraint vertex v_0 seen from s.) In this case, if there exists a new maximal constraint vertex w, w has to be a constraint vertex just discovered, for otherwise the polygon has empty kernel. Moreover, w belongs to the quadrant D_u of u and hence the clockwise angle $\widehat{suw}$ does not exceed $\pi/2$. Therefore, the robot at u lies on or outside the semicircle with diameter sw (Fact 2.1), and it will try to walk along the line su away from s in an attempt to see the cave next to w.

The above two cases do not take into account the fact that the robot may take advantage of what it has seen in order to avoid wandering around the polygon P. Clearly, the kernel of P is a subset of the inner halfplanes of the edges of P and of the inner halfplanes of the constraint vertices. Since the robot seeks to locate the kernel, it seems reasonable that it should not leave the inner halfplane of any of the polygon edges or constraint vertices which it sees or has seen. To be able to do that, the robot maintains the *free polygon* which is the subset of P in which the robot may walk (the free polygon is a combination of the gaining and keeping wedges introduced in [9] and also used in [13] for the same purpose). Initially, the free polygon is the intersection of the inner halfplanes of the visible edges and the visible constraint vertices from the starting point s. As a new edge or a new constraint vertex becomes visible, the robot updates its free polygon by intersecting it with the corresponding inner halfplane.

By requiring that the robot maintains the free polygon up to date and remains in it, we ensure that the portion of the polygon seen by the robot never decreases; at the same time, the free polygon keeps shrinking and when the robot reaches the kernel, the free polygon is precisely the kernel of P. Additionally, a left (resp., right) constraint vertex will remain so until both edges incident upon it become visible; it will not turn into a right (resp., left) constraint vertex, which might happen if the robot zig-zagged inside P.

At any time during its trip, the robot lies at a point, say, p, on the boundary of the current free polygon and it can only walk in the free polygon, that is, in the wedge delimited by the lines supporting the free polygon edges that are incident upon p. Since the free polygon is defined as the intersection of halfplanes, the opening angle of this wedge does not exceed π. Because the line supporting the edge to the left of p (with respect to the robot's motion towards the interior of the free polygon) bounds the current free polygon from the left, we call it a *left-bounding* line; similarly, the line supporting the edge to the right of p is a *right-bounding* line.

The following two cases complete the path planning strategy of the robot.

3. *The robot's intended course leads or lies outside the free polygon.* Then the robot walks along the boundary of the free polygon as close to the intended course as possible (Figure 7). In particular, let v be the currently maximal constraint vertex and p' be a point on the intended course of the robot that falls outside the free polygon. Then, the line vp' intersects the boundary of the free polygon in at most two points; the point p closest to p' is where the robot should be. (Note that if vp' does not intersect the free polygon, then the kernel of P is empty, and the robot stops and reports it.) In terms of left- and right-bounding lines, the robot walks along the left-bounding (right-bounding, respectively) line of the current free polygon if and only if the intended course leads to the left (right, respectively) of the free polygon.

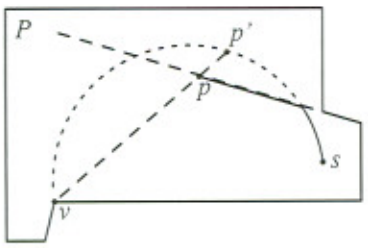


Figure 7

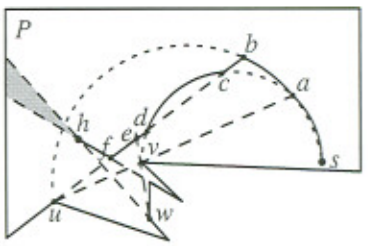


Figure 8

4. *An edge that was not visible becomes visible.* Then, the robot updates the free polygon by intersecting it with the inner halfplane of that edge. Note that this case has to be executed in case 2.

An example is shown in Figure 8: The robot starts along the semicircle with diameter sv. At a it sees the new maximal constraint vertex u and it starts walking along the semicircle with diameter su. At b, it sees the cave next to u; the new maximal constraint vertex is v again and so the robot walks along the line supporting bu. When it reaches the semicircle with diameter sv at point c, its starts walking along the semicircle. At d, it has to follow the left-bounding line bu. At e, it detects the new maximal constraint vertex w; the semicircle with diameter sw lies outside (and to the left of) the free polygon, so the robot continues walking along the left-bounding line bu, and later at f along the left-bounding line fh. At h, it sees the cave next to w and all the other left caves, and it stops.

It is important to note that the ending point h lies on the line supporting the edge which was seen last. Another important observation pertains to the way the value of the angle $\widehat{psv_0}$ behaves, where p denotes the current position of the robot on its way from s to h, and v_0 is the maximal left constraint vertex as observed from s. In the most general case, the following behavior of the angle $\widehat{psv_0}$ is exhibited: it is initially $\pi/2$, then it decreases, potentially reaching 0 but not decreasing below 0 (sub-path from s to f in Figure 8), and then it increases (sub-path from f to h). (Note that the robot may walk along sv_0.) However, two special cases may arise: first, the value of $\widehat{psv_0}$ is always decreasing from s to h (for example, consider the case that the caves of both v and w of Figure 8 were visible at f), and second, the value of $\widehat{psv_0}$ is always non-decreasing. The latter case may occur if, due to clipping, the left-bounding line of the free polygon is farther to the right from the semicircle with diameter sv_0; in this case, the robot will not follow any of the semicircles defined by s and the maximal left constraint vertices.

Lemma 3.1. *Suppose that the angle $\widehat{psv_0}$ decreases and then increases, reaching its minimum value when the robot is at the point x. Then,*

(i) x is either on or outside the corresponding semicircle,

(ii) the part of the robot's path past x lies outside the semicircle defined by s and the currently maximal constraint vertex.

Proof: (i) Consider a line l through s infinitesimally to the left of x. The line intersects the robot's path at two points q and q' and let q be closer to s than q'; then, q belongs to the robot's path from s to x, whereas q' belongs to the robot's path past x. Moreover, all the points of the line segment qq' belong to the free polygon. If x were inside the corresponding semicircle, then so would q as well as any point r of the segment qq' in a sufficiently small neighborhood of q. This is a contradiction, however, since the robot tries to stay as close to the corresponding semicircle as the free polygon allows it; therefore, on its way from s to x, it would not have gone through q if it were free to be at r or even closer to the semicircle.

(ii) Let v be the currently maximal constraint vertex when the robot is at x. Then, because of (i), the angle $\widehat{sxv} \leq \pi/2$ (Fact 2.1). Since x is the point where the angle $\widehat{psv_0}$ reaches its minimum value, then the course of the robot past x lies on or to the right of the directed line $\vec{sx}$. Moreover, it lies in the inner halfplane of v at x, that is, to the right of the directed line $\vec{xv}$ (Figure 9). On the other hand, if w is the maximal constraint vertex when the robot is at p (no matter whether this is a newly seen constraint vertex or one that has been seen before the robot reached x), w has to lie on or to the left of $\vec{sx}$ (for, otherwise, the line sw would be a clipping line and x would never have been reached), and on or to the left of $\vec{xv}$ (for, otherwise, v would not be maximal when the robot was at x). Thus, for any point p of the robot's path past x, the angle $\widehat{spw}$ is less than $\widehat{sxv}$. To see this, let q be the point of intersection of the lines pw and sx: if q belongs to the line segment sx, the above statement is clearly true given that w is on or to the left of $\vec{xv}$; if q does not belong to the line segment sx, $\widehat{spw} \leq \widehat{sqw} \leq \widehat{sxv}$ (with equality holding only if $p = x$). Therefore, $\widehat{spw} < \pi/2$, and x lies outside the semicircle with diameter sw. ∎

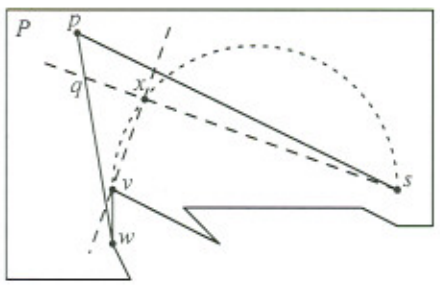


Figure 9

3.2 The general case. Our strategy for the general case consists of applying the one-sided strategy twice, once for the left caves and once for the right caves. In particular, the robot starts by applying the one-sided strategy for the left caves until either all the left caves are seen or it determines that the kernel is empty. In the latter case, it aborts its mission and reports its finding. In the former case, it stops when it entirely sees the last left cave. Then, it turns its attention to the right caves. If all the right caves have become visible, then the robot stops because it has reached the kernel. If there still exist right caves, the robot applies the one-sided case once again for the right caves this time.

Suppose that the robot is at point h, when it finally sees all the left caves. Then, the robot finds the maximal right constraint vertex u and updates its free polygon by intersecting it with the inner halfplane of u at h. The robot's intention is to walk along the semicircle C_{su} with diameter su; however, it has to reach C_{su} first. To do this, the robot tries to walk along the line hu towards the semicircle; by walking in this direction, the robot does neither gain nor lose visibility of the cave next to u. Of course, this course is subject to clipping about the free polygon; so, if the path along hu towards C_{su} leads outside the free polygon, the robot follows left-bounding lines if h is inside C_{su} and right-bounding lines if h is outside C_{su}. The four main cases are shown in Figures 10 and 11 depending on whether h is before or after the point x where the angle $\widehat{psv_0}$ achieves its minimum value. In Figure 10, h is before or coincides with x: in the case (a), h is inside C_{su} and the robot walks along the left-bounding line, i.e., along hv towards v; in the case (b), h is outside C_{su} and the robot walks along the right-bounding line, i.e., along hv away from v this time. In Figure 11, h is after x: in the case (a), h is inside C_{su} and the robot walks along the left-bounding line of the free polygon at h; in the case (b), h is outside C_{su} and the robot walks along the right-bounding line, i.e., along hv away from v.

It is interesting to observe that if h is after x and h is outside C_{su}, the maximal right constraint vertex u does not fall in the inner halfplane of hv (Figure 11(b)). This follows from the fact that $\widehat{vhs} < \pi/2$ (since h is outside C_{sv}), and that $\widehat{shu} < \pi/2$ (since h is outside C_{su}).

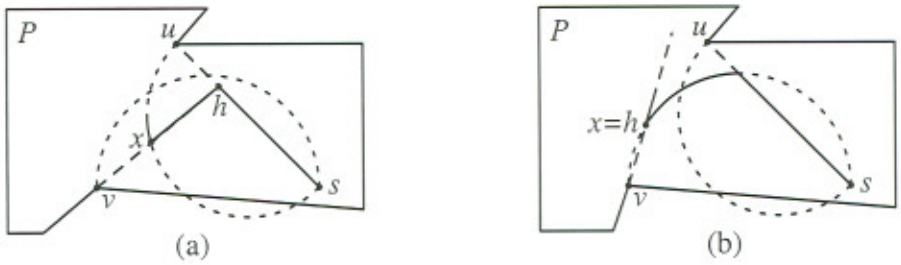


Figure 10

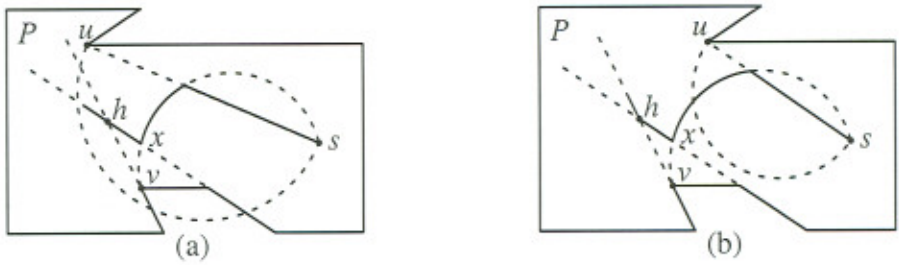


Figure 11

The final path consists of two sub-paths, one from s to h and the other from h to the final point t, each similar to the path shown in Figure 8. That is, each one of them consists of a number of clipped circular arcs and line segments (cases 1 and 2 of Section 3.1), potentially followed by one or more line segments that result from clipping whenever the corresponding semicircles fall outside the free polygon. Our observation in Section 3.1 about the behavior of the values of the angle $\widehat{psv_0}$ (where p is the robot's current position and v_0 is the maximal left constraint vertex as observed from s) is generalized and implies that, in the most general case, $\widehat{psv_0}$ is initially $\pi/2$, then decreases, potentially reaching 0 but not decreasing below 0, then it starts increasing assuming values up to $\widehat{u_0sv_0}$ (where u_0 is the maximal right constraint vertex as observed from s), and then it may start decreasing again up to 0.

3.3 Simulating the strategy. The obvious way to simulate a motion strategy involves starting at the predetermined starting position and executing small steps applying the rules of the strategy. This method has the obvious disadvantage that a good approximation of the robot's path requires a large number of steps which may lead to increased execution time and large errors resulting from accumulated numerical errors at each step.

A second approach is to split the given polygon P into regions in each of which the robot follows the same curve. Clearly, we will have to split P about the lines supporting the polygon edges incident upon reflex vertices. Moreover, we need to split P about lines that connect pairs of (left or right) constraint vertices that consecutively become maximal. To do that, we find the tree of shortest paths inside P from s to all the reflex vertices and we split P about the lines supporting the edges of this tree as well. Then, the robot can traverse any of the resulting regions in one step. This method involves fewer steps compared to the previous one but it requires computing the partition of the polygon about the above mentioned lines, whose total number is linear in the number n of polygon vertices. Building the partition requires $O(n^2)$ space and it can be done incrementally in $O(n^2)$ time in a fashion similar to the incremental construction of an arrangement of lines [7, 6]. The free polygon is maintained by turning on or off a bit associated with each region, while the only

computation in each region involves finding the points of intersection of the path with the region boundary.

4. Competitiveness Analysis.

In order to compute the competitive factor of our strategy, we need to compute the worst-case ratio of the length of the path resulting from the application of our strategy over the length of the line segment connecting the starting point s to the ending point t. Obviously, the worst case scenario involves double application of the one-sided case. Moreover, in order to simplify our analysis, we will consider an "augmented" path by ignoring (most of) the clipping while making sure that (i) the length of the augmented path is no less than the length of the actual path and (ii) the augmented path can be partitioned into pieces for which a bound on the competitive factor can be easily derived.

4.1. L-path and r-path. Before we describe the "augmentation" procedure, we review the important stops in the robot's path and define the l-path and r-path which will be used to augment the path. The robot first applies the one-sided strategy trying to see all the left caves; let h be the final point during this phase, that is, the point from where all the left caves are visible. Then, the robot applies the one-sided strategy again, for the right caves this time. As mentioned in Section 3.2, the angle $\widehat{psv_0}$ (defined by the current position p of the robot, the starting position s, and the maximal left constraint vertex v_0 observed from s) decreases, then it may increase and finally it may decrease again; let x and y be the turning points where these changes of monotonicity occur (if the robot walks along the line sx or sy, we let x and y be the closest such points to s). Note that x may coincide with h or may be before or after h along the robot's path; y may coincide with t, although this is not true in the most general case. Clearly, the sub-path from s to y lies in the closed halfplane to the left of the directed line $\vec{sx}$; similarly, the sub-path from x to t lies in the closed halfplane to the right of $\vec{sy}$.

Moreover, as mentioned earlier, the point h lies on the line supporting the polygon edge that just became visible at h; let l_h be that line. Then, l_h is a right-bounding line of the free polygon at h. Similarly, the ending point t lies on the line l_t supporting the edge that became visible last, and l_t is a left-bounding line of the free polygon at t.

We define the *l-path* as the path that the robot would follow if it only applied cases 1 and 2 of Section 3.1 from its starting position s until it either saw all the left caves or reached the line sx, whichever came first; in the former case, we extend the l-path by adding a line segment from the final point to the point of intersection of sx with the left-bounding line of the free polygon at the final point. Because clipping is ignored, this left-bounding line supports a polygon edge next to a maximal left constraint vertex; this edge is not necessarily the edge that became visible last. As a summary, the l-path consists of a sequence of circular arcs (case 1 of Section 3.1) occasionally separated by a line segment along a line supporting an initially invisible polygon edge (case 2 of Section 3.1). We define the *r-path* similarly: this is the path that the robot would follow if it only applied cases 1 and 2 of Section 3.1 starting from s until it either saw all the right caves or reached the line sy; again, if the robot has seen all the right caves before it reached the line sy, we extend the r-path accordingly. We finally define the *l-region* as the closed region bounded by the l-path and the line sx; similarly, the *r-region* is the closed region bounded by the r-path and the line sy.

For the l- and r-path, the following lemmata hold, which will be important in establishing that the l- and r-path are curves with increasing chords.

Lemma 4.1. *For any point p of the l-path, the part of the l-path from the starting point s up to p belongs to the closed quadrant A_p of p, whereas the part of the l-path past p belongs to the quadrant C_p of p.*

Proof: One needs to consider the different cases for p: on a circular arc, at the intersection of two arcs, at the intersection of an arc and a line segment, on a line segment. The lemma follows from the fact that for any point q of a semicircle with diameter ab, the angle $\widehat{aqb}$ is equal to $\pi/2$ (see Fact 2.1). (Figure 12 gives some examples for illustration purposes; the crosses indicate the lines delimiting the quadrants.) ∎

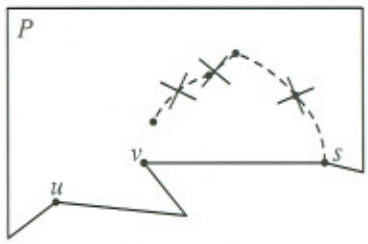


Figure 12

A similar lemma holds for the r-path.

Lemma 4.2. *For any point p of the r-path, the part of the r-path from the starting point s up to p belongs to the closed quadrant D_p of p, whereas the part of the r-path past p belongs to the quadrant B_p of p.*

The above lemmas and the fact that the l- and r-paths are defined up to the point of intersection with the lines sx and sy respectively imply the following corollary.

Corollary 4.1. *The boundaries of the l- and the r-region are intersected by any line through the starting position s in at most one point other than s.*

We close this section with another useful observation.

Observation 4.1. *The point h from which all the left caves are finally visible does not belong to the interior of the l-region. Similarly, the final point t from which the entire polygon is visible does not belong to the interior of the r-region.*

Proof: Clearly, h belongs to the intersection H of the inner halfplanes of all the invisible edges incident upon the left constraint vertices. We will show that the interior of the l-region belongs to the complement of H. If the l-path ends with a circular arc associated with a left constraint vertex, say, v, then the entire l-region belongs to the complement of the inner halfplane of the invisible edge incident upon v. If the l-path ends with a line segment (case 2 of Section 3.1, or extension of the l-path to reach the line sx), then this line segment lies on a line supporting an invisible edge; again, the interior of the l-region belongs to the complement of the inner halfplane of that edge. The case for t is similar. ∎

4.2. Augmenting the robot's path. The robot tries to follow the l-path and the r-path if possible, or otherwise stay as close to them as possible. On its course from the starting point s to h (the case is similar for the part from h to the ending point t), it follows (parts of) the l-path, may move outside the l-region due to clipping about a left-bounding line (when the l-path leads farther left than the left boundary of the free polygon), or may move inside the l-region due to clipping about a right-bounding line (when the l-path leads farther right than the right boundary of the free polygon). In general, the robot may move in and out of the l-region several times; after it has moved in, it may walk along several different right-bounding lines (tracing a convex curve inside the l-region), whereas after it has moved out, it may follow several different left-bounding lines (tracing a concave curve outside the l-region). It is important to observe:

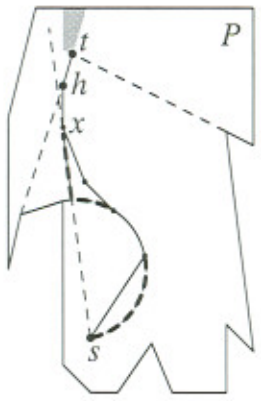


Figure 13

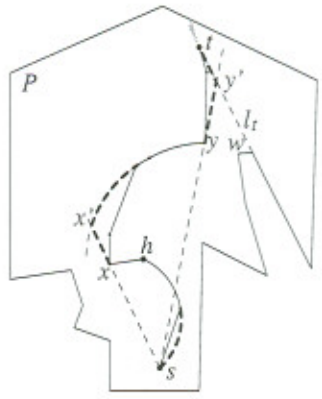


Figure 14

Observation 4.2. *The robot never follows a left-bounding line followed by a right-bounding line (or vice versa), except at the point h where it has seen all the left caves and sets off to see the right caves.*

The observation follows from the fact that the robot tries to stay as close to the corresponding semicircle as it can and if this is farther left (right, respectively) than the left (right, respectively) boundary of the free polygon, the robot will keep following the left (right, respectively) boundary of the free polygon until it reaches it, if ever.

Now we are ready to see how the actual robot's path is being augmented; we will also define the points x' and y' which will be crucial in partitioning the augmented path into curves with increasing chords. We concentrate on the most general case in which $x \neq s$ (i.e., the angle $\widehat{psv_0}$ starts by decreasing) and $x \neq t$; the special case where $x = s$ and the two special cases where $x = t$ are investigated in Section 4.4. Note that y may or may not coincide with t.

1. *the part of the robot's path from s to x:* We recall that x may be either on the l-path or outside the l-region; in the latter case and if additionally h coincides with or is reached after x, then the robot has been walking along left-bounding lines from the last point of its course on the l-path up to x. Recall also that h is either on the l-path or outside the l-region (Observation 4.1); if it is outside the l-region, then again the robot has been walking along left-bounding lines. In all cases where the robot walks along left-bounding lines after it leaves the l-region (no matter whether h is reached before or after x), the sub-path from s to x is augmented by considering the entire l-path, followed by a line segment from the final point of the l-path to x along sx (Figure 13); this includes as a special case the case where x belongs to the l-path. It remains to consider the cases where the robot walks along right-bounding lines. There are two cases to consider. In the first case, h belongs to the l-path, x is reached after h, and the robot walks along a right-bounding line past h towards x; this implies that $x = t$ and is the special case 2 which we consider in Section 4.4. In the second case, h is outside the l-region, x is reached after h, and the robot walks along a right-bounding line past h towards x; this again implies that $x = t$ and is the special case 3 considered in Section 4.4.
2. *the part of the robot's path from x to y:* We distinguish two cases depending on whether the robot walks along left- or right-bounding lines past x.

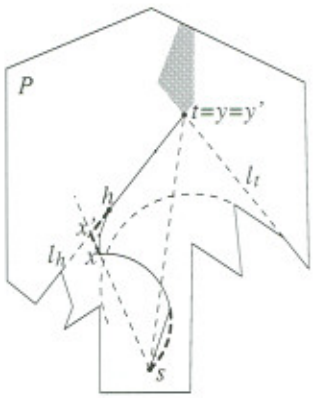


Figure 15

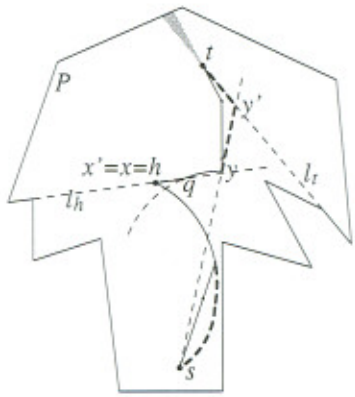


Figure 16

(i) the robot walks along a left-bounding line past x. If the point h is before x or coincides with x, then x must belong to the r-region for the robot to follow a left-bounding line past x. We let x' be the point of intersection of sx with the r-path, and we augment the sub-path from x to y by considering the line segment xx', followed by the r-path up to its intersection with the line sy, followed by the line segment from that point to y (Figure 14). If the point h is after x, then past h the robot may walk along a left- or a right-bounding line depending on whether h belongs to the r-region or not. Let q be the point of intersection of the lines sx and l_h. If h is outside the r-region, or if h belongs to the r-region but q does not, we set $x' = q$ and we augment the path by considering the line segment xx' (along sx), followed by a line segment along l_h from x' to the point of intersection with the r-path, followed by a line segment from that point to y along sy (Figure 15). If both q and h belong to the r-region, then we let x' be the point of intersection of the line sx with the r-path, and the sub-path from x to y is augmented by considering the line segment xx' (along sx), followed by the r-path from x' to its final point on the line sy, followed by the line segment from that point to y (along sy); the situation is similar to the one depicted in Figure 14.

(ii) the robot walks along a right-bounding line past x. Then, h cannot be before x, for, if h were reached before x, the robot must have been walking along right-bounding lines from h to x; this implies that $x = t$, a contradiction to the continuation of the path past x. Moreover, h cannot be after x either; if h were reached after x, then h would be outside the l-region and the robot would be walking along left-bounding lines from x to h. Therefore, $h = x$, and we set $x' = h$. Additionally, h lies outside the r-region (otherwise, the robot would not be following a right-bounding line past x). Let q be the point of intersection of l_h with the r-path (if l_h intersects a line segment of the r-path, then q is the point of intersection of l_h with the immediately following semicircle); if the line l_h does not intersect the r-path, we let q be the point of intersection of l_h and sy. Then, the sub-path from x to y is augmented by considering the line segment xq along l_h, potentially followed by the r-path from q to its intersection with the line sy (if q does not belong to the line sy), followed by the line segment from that point to y.

3. *the part of the robot's path from y to the final point t:* If $y = t$, then we set $y' = y = t$. If $y \neq t$, then the path past y lies outside the corresponding semicircles (Lemma 3.1) and the robot on its way to t walks along right-bounding lines only. So, this part of the actual path is augmented by considering the polygonal line formed by the segments yy' and $y't$, where y' is the point of intersection of the lines sy and l_t (Figure 16).

It is important to observe that the augmented path does not cross itself. This follows from the fact that x lies on the boundary or outside the l-region, y similarly lies on the boundary or outside the r-region and y lies in the inner halfplane of l_h. Moreover, the augmented path always moves along or to the left of the left-bounding lines that the robot follows and along or to the right of the right-bounding lines, thus enclosing the actual robot's path. Therefore, we have the following two observations.

Observation 4.3. *The path traveled by the robot and the augmented path have the same endpoints.*

Observation 4.4. *The path traveled by the robot can be produced by clipping the augmented path about the edges of a (shrinking) convex polygon.*

4.3. The competitive factor. With respect to the points x' and y', the augmented path can be seen as the concatenation of three sub-paths, one from s to x', one from x' to y', and one from y' to the final point t. The sub-path from s to x' consists of circular arcs (case 1 of Section 3.1) occasionally separated by a line segment along a line supporting an initially invisible polygon edge (case 2 of Section 3.1), potentially ending with a line segment along the line sx. The sub-path from x' to y' consists mainly of arcs and line segments (in accordance with cases 1 and 2 of Section 3.1) as well, but may begin with a line segment along l_h, and may end with a line segment along the line sy; the sub-path may degenerate into a two-segment polygonal line, one along l_h and the other along sy. Finally, the sub-path from y' to t is simply a line segment. See Figures 14-16. More importantly, the following lemmata hold.

Lemma 4.3. *The counterclockwise angle $\widehat{sx'y'}$ is at least equal to $\pi/2$.*

Proof: The definition of the point x' in the case 2 of the preceding section suggests that we need to consider two cases.

(i) x' is the point of intersection of sx with the r-path (if x is inside or on the boundary of the r-region): Then, x' lies on the semicircle of the currently maximal right constraint vertex (case 1 of Section 3.1), or on the line supporting an edge incident upon a right-constraint vertex which was initially invisible and became visible (case 2 of Section 3.1); in the latter case, x' lies inside the semicircle associated with the right constraint vertex. In either case, if w is the right constraint vertex, then the angle $\widehat{sx'w}$ is at least equal to $\pi/2$. Moreover, the point y and (a fortiori) the point y' lie on or to the left of the directed line $\vec{x'w}$ (see Figure 14). Therefore, $\widehat{sx'y'} \geq \widehat{sx'w}$, and the lemma follows.

(ii) x' is the point of intersection of sx with l_h: This case occurs if h is reached after x, or $x' = h$ and h lies outside the r-region. In either case, x' coincides with or is farther away from s than x; since x is on the boundary or outside the l-region (Lemma 3.1), x' lies on or outside the semicircle of the last maximal left-constraint vertex, say, v. Then, $\widehat{sx'v} \leq \pi/2$ (Fact 2.1). The lemma follows from the fact that y and (a fortiori) y' belong to the inner halfplane of l_h and thus the angle $\widehat{sx'y'}$ is at least equal to $\pi - \widehat{sx'v}$ (see Figure 15). ∎

Similarly,

Lemma 4.4. *If $y \neq t$, the clockwise angle $\widehat{sy't}$ is at least equal to $\pi/2$.*

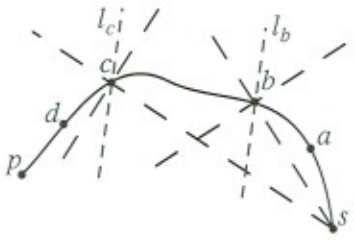


Figure 17

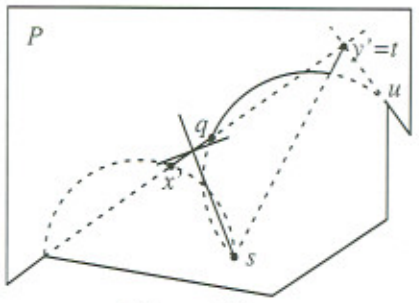


Figure 18

Proof: In accordance with Lemma 3.1, the point y lies on or outside the semicircle of the maximal right constraint vertex at t; let w be that maximal vertex. Then, y' has to lie on or outside the semicircle as well, as it lies farther away from s than y along sy and the line sy intersects the semicircle with diameter sw in at most one point other than s. Hence, $\widehat{sy'w} \leq \pi/2$. The lemma follows, since $\widehat{sy't} = \pi - \widehat{sy'w}$ (see Figure 14). ∎

Lemma 4.5. *The sub-path of the augmented path from s to x' is a curve with increasing chords.*

Proof: We first prove that for any point p of this sub-path, the part of the augmented path from s to p belongs to the closed quadrant A_p of p, while the part of the path from p to x' belongs to the closed quadrant C_p of p. This follows from Lemma 4.1 and the fact that it holds for all the points of the segment xx', which is along sx. Next, we consider 4 points a, b, c and d in that order along the augmented path. We draw the corresponding quadrants for the points b and c and draw the two lines l_b and l_c perpendicular to bc that pass by b and c respectively (Figure 17). Since c belongs to the quadrant C_b of b, l_b lies in the closure of the wedge defined by the quadrants B_b and D_b of b. Similarly, since b belongs to the quadrant A_c of c, l_c lies in the closure of the wedge defined by the quadrants B_c and D_c of c. Moreover, the point a lies in the quadrant A_b of b, that is, to the left of l_b. Similarly, the point d lies in the quadrant C_c of c, that is, to the right of l_c. Therefore, the length of ad is no less than the perpendicular distance of l_b and l_c, which by construction is equal to bc. ∎

Similarly,

Lemma 4.6. *The sub-path of the augmented path from x' to y' is a curve with increasing chords.*

Proof: This sub-path is similar to the sub-path from s to x' (and the proof is symmetric) with the exception that it may start with a line segment along the line l_h, and that it may degenerate into a two-segment polygonal line, one along l_h and the other along sy. Let v be the left constraint vertex associated with l_h; recall that x' lies on or outside the semicircle with diameter sv and thus we have that $\widehat{sx'v} \leq \pi/2$ (see case (ii) of Lemma 4.3). This implies that, in the case that the sub-path degenerates into a polygonal line consisting of two segments $x'q$ and qy', the angle $\widehat{x'qy'}$ is at least equal to $\pi/2$; from the triangle with vertices s, x', and q, we can see that $\widehat{x'qy'} > \widehat{qx's} = \pi - \widehat{sx'v}$. In a fashion similar to that used in the proof of the previous lemma, we can easily show that two line segments at an angle at least equal to $\pi/2$ form a curve with increasing chords. Let us now turn to the case that the sub-path starts with a line segment (along l_h), say, $x'q$, and is followed by a sub-path of the r-path. This may occur in both cases 2(i) and 2(ii) of Section 4.2; in either

case, x' is outside the r-region. Moreover, q belongs to an arc of the semicircle defined by s and a right constraint vertex; let that constraint vertex be u. Since x' is outside the semicircle with diameter su, the angle $\widehat{sx'u}$ is less than $\pi/2$, which implies that the line l_h does not intersect the line segment su, because $\widehat{sx'q} = \pi - \widehat{sx'u} > \pi/2$. The situation is therefore as shown in Figure 18, and it is easy to show that for any point p of $x'q$, the part of the path from x' to p belongs to the closed quadrant D_p of p, whereas the part of the path from p to y' belongs to the closed quadrant B_p of p. The lemma follows from a proof method similar to that used in the previous lemma, in light of Lemma 4.2 and the above observation for the line segment $x'q$. ∎

From the above, we conclude

Theorem 4.1. *Our strategy has a competitive factor of* $\sqrt{2(2\pi/3)^2 + 1} \simeq 3.126$.

Proof: Clearly, the length of the actual path traveled by the robot is no more than the length of the augmented path, as clipping with convex polygonal lines or curves leads to reduced path length (Observation 4.4 and Lemma 2.4). So, we only need to find an upper bound for the ratio of the length of the augmented path over the length of the line segment st, in order to have an upper bound for the competitive factor that we seek. Figure 19 shows the skeleton of the augmented path in the worst case; as stated in the Lemmata 4.3 and 4.4, the angles $\alpha = \widehat{sx'y'}$ and $\beta = \widehat{sy't}$ are at least equal to $\pi/2$.

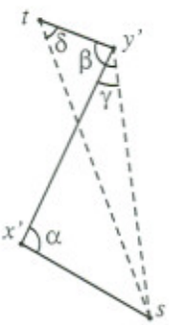


Figure 19

Let us denote by $|\widetilde{pq}|$ the length of the path from p to q as opposed to $|pq|$ which denotes the length of the line segment pq. Then the competitive factor r is

$$r = \frac{|\widetilde{sx'}| + |\widetilde{x'y'}| + |y't|}{|st|} \leq \frac{\frac{2\pi}{3}|sx'| + \frac{2\pi}{3}|x'y'| + |y't|}{|st|},$$

since the augmented sub-paths from s to x' and from x' to y' are curves with increasing chords (Lemmata 4.5 and 4.6) and therefore their lengths are not more than $2\pi/3$ times the lengths of the line segments sx' and $x'y'$ respectively (Lemma 2.3). If we apply the law of sines in the triangles $sx'y'$ and $sy't$ and factor out the length $|sy'|$ we find

$$r \leq \frac{\frac{2\pi}{3}\,\frac{\sin\gamma + \sin(\alpha+\gamma)}{\sin\alpha} + \frac{\sin(\beta+\delta)}{\sin\delta}}{\frac{\sin\beta}{\sin\delta}},$$

where $\pi/2 \leq \alpha < \pi$, $\pi/2 \leq \beta < \pi$, $0 < \gamma < \pi - \alpha$ and $0 < \delta < \pi - \beta$. Obviously, the right-hand side is maximized for the values of α and γ that maximize the term $\frac{\sin\gamma + \sin(\alpha+\gamma)}{\sin\alpha}$. The study of the behavior of the values of this term (by means of its partial derivatives in terms of α and γ) shows that the maximum value is achieved for $\alpha = \pi/2$ and $\gamma = \pi/4$, in which case

$$r \leq \frac{\frac{2\pi}{3}\,\sqrt{2} + \frac{\sin(\beta+\delta)}{\sin\delta}}{\frac{\sin\beta}{\sin\delta}}.$$

Maximizing the right-hand side over all the allowed values of β and δ yields $\sqrt{2(2\pi/3)^2 + 1} \simeq 3.126$, which is an upper bound for r. ∎

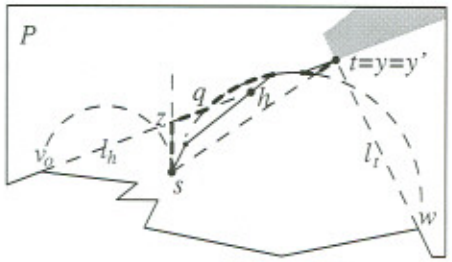


Figure 20

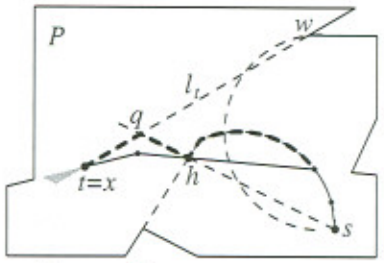


Figure 21

4.4. Special cases in the augmentation of the robot's path. In Section 4.2, we focused on the most general case for the robot's path. We will consider the special cases in this section.

1. The point x coincides with the starting point s.
Then the left-bounding line of the free polygon at s is farther right than the semicircle with diameter sv_0, where v_0 is the maximal left constraint vertex as observed from the starting position s. Therefore, the robot keeps walking along left-bounding lines up to the point h, at which point it may continue by walking along a circular arc if h belongs to the r-path, a left-bounding line if h is inside the r-region, or a right-bounding line (the line l_h or a clipping line towards the inner halfplane of l_h) if h is outside the r-region. In any case, let z be the point of intersection of l_h with the line perpendicular to sv_0 at s (Figure 20). Furthermore, let q be the point of intersection of the line l_h with the r-path (if l_h intersects a line segment of the r-path, then q is the point of intersection of l_h with the immediately following semicircle); if the line l_h does not intersect the r-path, we let q be the point of intersection of l_h and sy. If z belongs to the r-region, then the robot's path from s to y is augmented by considering the r-path from s to its intersection with the line sy, followed by the line segment from that point to y. If z is outside the r-region (no matter whether h does or does not), the robot's path from s to y is augmented by considering the line segment sz, followed by the line segment zq along l_h, potentially followed by the r-path from q to its intersection with the line sy (if q does not belong to the line sy), followed by the line segment from that point to y. (This case is similar to the case 2(ii) of Section 4.2.) If $y \neq t$, then the point y' is defined as described in the case 3 of Section 4.2, and the path is augmented accordingly.

In the former case, the augmented path consists of the part of the r-path from s to y', followed by the line segment $y't$ along l_t. Since $\widehat{sy't} > \pi/2$ (as in Lemma 4.4), the method used in the proof of Theorem 4.1 yields a competitive factor of $\sqrt{(2\pi/3)^2 + 1} \simeq 2.32$. In the latter case, the augmented path is the concatenation of the line segment sz, followed by the line segment zq along l_h, followed by the r-path from q to its point of intersection with the line sy, followed by the line segment from that point to y', followed by the line segment $y't$ (Figure 20). By construction, z is outside the l-region; then, the angle $\widehat{szq}$ is larger than $\pi/2$, implying that $\widehat{szy'} > \pi/2$ as well. Moreover, $\widehat{sy't} \geq \pi/2$. The point z is outside the r-region too; thus, if w is any maximal right constraint vertex encountered on the r-path past q, the angle $\widehat{sqw} \leq \pi/2$; given that $\widehat{szq} > \pi/2$, we conclude that the line l_h does not intersect the line segment sw. That is, the sub-path of the augmented path from z to y' is exactly like the sub-path $x'y'$ of the augmented path in the general case, and is therefore a curve with increasing chords. Therefore, this case is similar to the general case that we have considered in Section 4.3, except that the first sub-path is just a line segment

and not a curve with increasing chords. Computing the competitive factor yields a value of $\sqrt{(2\pi/3)^2+2} \simeq 2.527$.

2. The point h lies on the l-path, the turning point x is reached after h, and the robot walks along right-bounding lines on its way from h to x.
Then, h is outside the r-region; this along with Observation 4.2 imply that the robot will keep walking along right-bounding lines until it reaches x and that $x = t$. If q is the point of intersection of the lines sh and l_t, the path from s to t is augmented by considering the l-path from s to h, followed by the line segment hq along sh, followed by the line segment qt along l_t (Figure 21).

This augmented path can be seen as the concatenation of the sub-path from s to q, which is a curve with increasing chords (Lemma 4.5), and the line segment qt. Additionally, q is outside the r-region, which implies that the angle $\widehat{sqt} = \pi - \widehat{sqw} > \pi/2$. This case is simpler than the general case we have considered and yields a competitive factor of $\sqrt{(2\pi/3)^2+1} \simeq 2.32$.

3. The point h is outside the l-region, the turning point x is reached after h, and the robot walks along a right-bounding line l past h towards x.
Then, the starting point s has to belong to the inner halfplane of l. On the other hand, because h is outside the l-region, the robot must have walked along left-bounding lines to reach h. Given that $x \neq h$, the only possibility is that the line l is both a left- and a right-bounding line; then x lies on l and $x = t$. The path from s to x is augmented by considering the entire l-path, followed by the line segment along sx from the final point of the l-path to x.

This augmented path is similar to the augmented sub-path from s to x', which is a curve with increasing chords (Lemma 4.5). Then, the competitive factor does not exceed $2\pi/3$.

5. Path Properties.

It is interesting to observe that every point of the robot's path belongs either to a semicircle defined by the starting point and a vertex of the polygon P (a maximal constraint vertex) or to the line supporting an edge of P. This guarantees that an actual robot following our strategy is not expected to deviate from the intended course, as opposed to other strategies where this is possible because the motion of the robot is dependent on the current position. Consider, for example, Icking and Klein's strategy where the robot follows the bisector of an angle with apex the current position; this has the drawback that, due to accumulated numerical error, the robot may deviate substantially from the expected course.

Additionally, the following lemmata establish that the path constructed by our strategy leads directly to the kernel point closest to the starting point and that it has size linear in the number of vertices of the polygon P.

Lemma 5.1. *The path resulting from the application of the above described strategy reaches the kernel of the polygon at the kernel's point that is closest to the starting point s.*

Proof: Let t be the point on the robot's course from where it first sees the entire P and let w be the last maximal constraint vertex. Then, t lies on the line l_t supporting the edge e of P which became visible last; this edge is incident upon w. If t lies on the semicircle with diameter sw (e.g., because the robot has been walking along the semicircle when it reached t), then clearly t is the point of the kernel that is closest to s because the line st is perpendicular to the edge of the kernel on the line l_t and the kernel is convex. However,

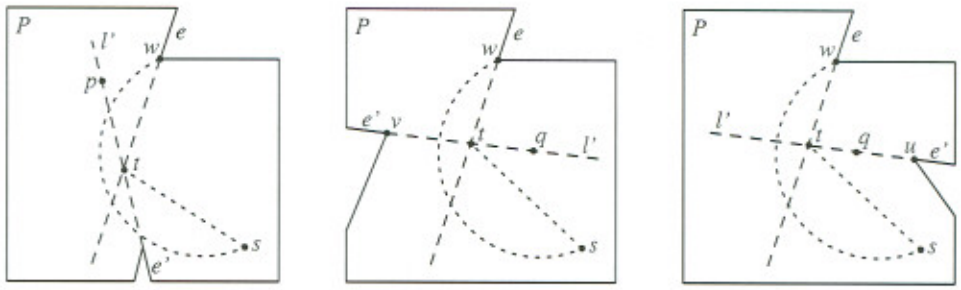


Figure 22

t may not lie on the semicircle; the robot may have been walking along a (clipping) line l' when it reached t. Because the robot is trying to stay as close to the semicircle as the clipping allows it, t is closer to s than any other point of the kernel edge on l_t. So, we need only show that t is closer to s than any other point of the kernel edge on l', or equivalently that the line perpendicular to l' that passes through s intersects l' at a point that belongs to the closure of the complement of the inner halfplane of e.

Let e' be the edge of P that causes the clipping along the line l', and let q be the first point of l' that the robot ever reached. We can show that the line perpendicular to l' that passes through s intersects l' at a point that belongs to the closure of the complement of the inner halfplane of e, by showing either that the angle $\widehat{sqt} \geq \pi/2$ or that the angle $\widehat{stp} \geq \pi/2$ for any point p of l' that belongs to the inner halfplane of e. We distinguish two main cases depending on whether the point t is inside or outside the semicircle with diameter sw. Suppose that the point t is inside the semicircle; then, $\widehat{stw} > \pi/2$ (Fact 2.1). There are three possibilities.

(i) s belongs to the inner halfplane of e' (Figure 22(a)). So is w. Then, for any point p of l' that belongs to the inner halfplane of e we have that $\widehat{stp} \geq \widehat{stw} > \pi/2$.

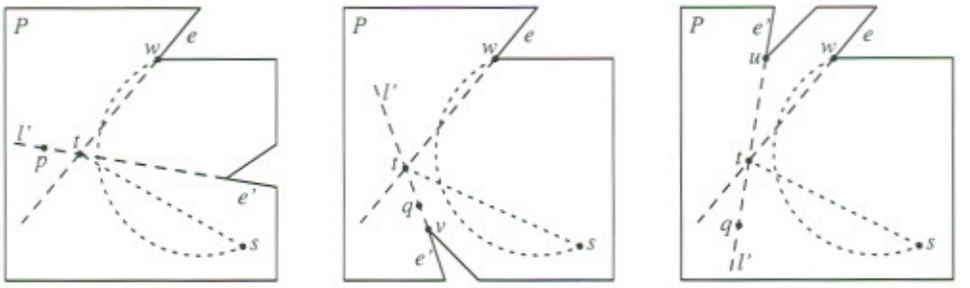
(ii) the edge e' is incident upon a left constraint vertex v (Figure 22(b)). Let v' be the maximal left constraint vertex just before the robot reached q; then, v' belongs to the inner halfplane of e'. Since the robot is free to walk from q towards t, q must be on or inside the semicircle defined by s and v'; this implies that $\widehat{sqv'} \geq \pi/2$. Moreover, it turns out that v' is precisely v; if $v' \neq v$, the line segment qt does not belong to the inner halfplane of v' at q and hence the robot would not be allowed to walk towards t. Therefore, $\widehat{sqt} = \widehat{sqv} = \widehat{sqv'} \geq \pi/2$.

(iii) the edge e' is incident upon a right constraint vertex u (Figure 22(c)). Let u' be the maximal right constraint vertex just before the robot reached q (note that u' may very well be u); then, u' belongs to the inner halfplane of e' and thus $\widehat{squ} \leq \widehat{squ'}$. Since the robot is free to walk from q towards t, q must be on or outside the semicircle defined by s and u'; this implies that $\widehat{squ'} \leq \pi/2$. Therefore, $\widehat{sqt} = \pi - \widehat{squ} \geq \pi - \widehat{squ'} \geq \pi/2$.

Suppose now that the point t is outside the semicircle with diameter sw; then, $\widehat{stw} < \pi/2$ (Fact 2.1). Again, there are three possibilities, similar to those of the previous case.

(i) s belongs to the inner halfplane of e' (Figure 23(a)). Then, w is not. Moreover, for any point p of l' that belongs to the inner halfplane of e we have $\widehat{stp} \geq \pi - \widehat{stw} > \pi/2$.

(ii) the edge e' is incident upon a left constraint vertex v (Figure 23(b)). Let v' be the maximal left constraint vertex just before the robot reached q (note that v' may very well be v); then, $\widehat{sqv} \leq \widehat{sqv'}$. Again, since the robot is free to walk from q towards t, q



(a) (b) (c)

Figure 23

must be on or outside the semicircle defined by s and v'; this implies that $\widehat{sqv'} \leq \pi/2$. Then, $\widehat{sqt} = \pi - \widehat{sqv} \geq \pi - \widehat{sqv'} \geq \pi/2$.

(iii) the edge e' is incident upon a right constraint vertex u (Figure 23(c)). Let u' be the maximal right constraint vertex just before the robot reached q and u' belongs to the inner halfplane of e'. Then, q must be on or inside the semicircle defined by s and u'; this implies that $\widehat{squ'} \geq \pi/2$. Moreover, it turns out that u' is precisely u; if $u' \neq u$, the line segment qt does not belong to the inner halfplane of u' at q and hence the robot would not be allowed to walk towards t. Therefore, $\widehat{sqt} = \widehat{squ} = \widehat{squ'} \geq \pi/2$. ∎

Lemma 5.2. *The path that the robot follows in accordance with our strategy consists of $O(n)$ line segments or circular arcs, where n is the number of vertices of the polygon P.*

Proof: We need only prove the lemma for the one-sided case, since the entire path consists of two applications of that case. Let z be the last point common to both the robot's path and the l-path; then the robot's path from s to z is a clipped version of the part of the l-path from s to z. The proof for the one-sided case proceeds in two steps. First, we derive an upper bound on the number of arcs and line segments of the l-path from s to z; the effect of clipping and the line segments of the robot's path past z are taken into account in the second step. The l-path from s to z consists of circular arcs (corresponding to the maximal constraint vertices) occasionally separated by a line segment along an initially invisible polygon edge (cases 1 and 2 of Section 3.1). The semicircle corresponding to a constraint vertex v may contribute more than one arc only if a new maximal constraint vertex u appears, u's cave later becomes entirely visible and v becomes the maximal constraint vertex again (see Figure 6). In this case, u ceases to be a constraint vertex, so the cost for the additional arc and the line segment between the semicircles of u and v can be charged to u. Since the number of constraint vertices is less than n, the unclipped path of the robot up to z consists of less than $n + 2 \times n = 3n$ circular arcs and line segments.

Let us now take into account the line segments of the path past z and the effect of clipping about the edges of the free polygon; we note that the line segments past z are also due to clipping about free polygon edges. Since the free polygon is convex and it keeps shrinking, any of its edges will contribute one line segment in the robot's path unless one or more circular arcs or l-path line segments interrupt it: the two possible cases are shown in Figure 24. In the case (a), the corresponding semicircles will not contribute any more arcs as their remaining portions lie outside the free polygon; therefore, the number of additional segments is no more than the number of constraint vertices. In the case (b), an edge of the free polygon may contribute several line segments while a clipped semicircle may contribute

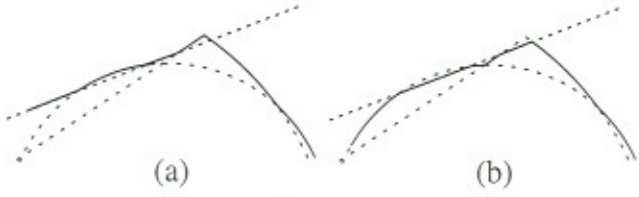


Figure 24

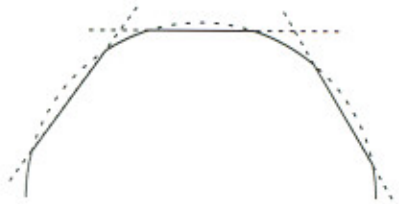


Figure 25

at most two arcs; however, every pair of consecutive line segments that the same edge contributes are separated by a vertex of the unclipped path of the robot (except for the starting point and z), which as we saw earlier consists of less than $3n$ circular and line segments and therefore has at most $3n$ vertices. Moreover, a semicircle may contribute several arcs due to clipping with the edges of the free polygon (Figure 25); for every additional arc, however, only the last clipping line may contribute more segments or arcs. So, we can account for all the additional segments that the clipping produces in all three cases, if we charge each edge of the free polygon with 2 units (one for a line segment and one for an additional arc) and each vertex of the unclipped path up to z with 2 units as well (one for the additional line segment and one for the additional arc), and add 2 more units for the last clipping line. In other words, the additional number of segments does not exceed $2 \times (n+2) + 2 \times 3n + 2$, since the total number of edges of the free polygon is at most $n+2$ (as mentioned earlier, they are contributed by the inner halfplanes of the edges of the polygon P and the inner halfplanes of the maximal constraint vertices detected from the starting point s.)

Summing up for both steps, we conclude that the total number of arcs and line segments in the entire path of the robot in the one-sided case is no more than $3n + 8n + 6 = O(n)$. ∎

6. Concluding Remarks – Open Problems.

We presented a strategy which enables a point robot to reach the point t of the kernel that is closest to the starting point s, and guarantees that the length of the path traveled is not longer than 3.126 times the length of the line segment st (that is, 3.126 times the shortest possible off-line path). The path generated consists of line segments and circular arcs whose total number is linear in the size of the polygon in which the robot moves. Our strategy has the interesting feature that the robot reaches the kernel at precisely the closest point t.

However, the above competitive factor cannot be guaranteed for negative instances, that is, when the polygon has empty kernel. In such cases, the ratio of the length of the path that our strategy imposes over the length of the shortest path which establishes that the kernel is empty may be unbounded. See, for example, Figure 26: our strategy entails following the semicircle with diameter sv and then moving along vz until we reach point z where it is found that the kernel is empty. The shortest off-line path involves visiting the vertices v and u. Therefore, as we arbitrarily scale the figure along the vertical axis, the competitive factor becomes arbitrarily large. This holds for all strategies where a point of the polygon seen by the robot never ceases to be in the robot's visible region thereafter (enforced by means of the free polygon in this work, and by means of the

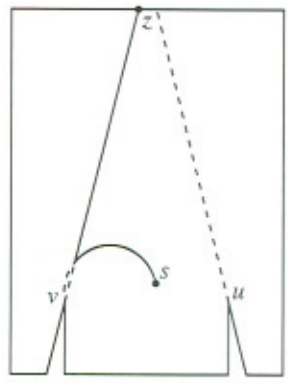


Figure 26

gaining and keeping wedges in [9] and [13]). Therefore, in order to improve on that, the strategy needs to relax this condition, as is done in [15]. Perhaps, a combination of methods may accommodate negative instances while improving the competitive factor of [15].

Experimental results suggest that the actual competitive factor may be smaller than the theoretical competitive factor of 3.126. More extensive experimentation is under way to investigate this possibility. If true, it would be interesting to come up with tighter theoretical bounds on the competitive factor of our strategy.

Of course, the ultimate open question is to invent strategies with smaller competitive factors which will close the gap between the current upper bound of $\sim$3.126 and the lower bound of $\sim$1.48. To this effect, perhaps ideas like the ones in [11] may be of help.

Finally, good or better competitive solutions are needed for other motion planning problems in unknown environments. A recent paper by López-Ortiz and Schuierer [15] has addressed two interesting problems in this class: finding out whether a given polygon is star-shaped (i.e., it has non-empty kernel), and locating a target (which will be recognized when seen) in a polygon with non-empty kernel. The currently best competitive factor for the first problem is 46.35. The currently best competitive factor for the second problem is 12.72 and is coupled with a lower bound of 9.

Acknowledgements. I would like to thank Bill Karaiskos whose program (implementing the strategy described in this paper) helped to produce several of the included figures.

7. References.

1. A. Blum, P. Raghavan, and B. Schieber, "Navigating in unfamiliar Geometric Terrain," *Proc. 23th Annual ACM Symposium on Theory of Computing* (1991), 494–504.
2. A. Datta, C. Hipke, and S. Schuierer, "Competitive searching in polygons – beyond generalized streets," *Proc. 6th International Symposium on Algorithms and Computation, LNCS 1004* (1995), 32–41.
3. A. Datta and C. Icking, "Competitive searching in a generalized street," *Proc. 10th ACM Symposium on Computational Geometry* (1994), 175–182.
4. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer-Verlag, 1997.
5. X. Deng, T. Kameda, and C. Papadimitriou, "How to learn an Unknown Environment," *Proc. 32nd IEEE Symposium on Foundations of Computer Science* (1991), 298–303.
6. H. Edelsbrunner and L.J. Guibas, "Topologically sweeping an arrangement," *Journal of Computer and Systems Science* 38 (1989), 165–194. Corrigendum in 42 (1991), 249–251.
7. H. Edelsbrunner, J. O'Rourke, and R. Seidel, "Constructing arrangements of lines and hyperplanes with applications," *SIAM Journal on Computing* 15(2) (1986), 341–363.
8. Y.K. Hwang and N. Ahuja, "Gross Motion Planning – A Survey," *ACM Computing Surveys*, Vol. 24, No. 3 (1992), 219–291.
9. C. Icking and R. Klein, "Searching for the Kernel of a Polygon — A Competitive Strategy," *Proc. 11th Annual ACM Symposium on Computational Geometry* (1995), 258–266.

10. C. Icking, R. Klein, and E. Langetepe, "An Optimal Competitive Strategy for Walking in Streets," *Technical Report*, FernUniversität Hagen, Praktische Informatik VI, Hagen, Germany, 1998.

11. C. Icking, R. Klein, and L. Ma, "An Optimal Competitive Strategy for Looking around a Corner," *Proc. 5th Canadian Conference on Computational Geometry* (1993), 443–448.

12. R. Klein, "Walking an Unknown Street with Bounded Detour," *Computational Geometry: Theory and Applications* 1 (1992), 325–351.

13. J.-H. Lee, C.-S. Shin, J.-H. Kim, S.Y. Shin, and K.-Y. Chwa, "New Competitive Strategies for Searching in Unknown Star-Shaped Polygons," *Proc. 13th Annual ACM Symposium on Computational Geometry* (1997), 427–429.

14. A. López-Ortiz and S. Schuierer, "Generalized Streets revisited," *Proc. 4th European Symposium on Algorithms, LNCS 1136* (1996), 546–558.

15. A. López-Ortiz and S. Schuierer, "Position-independent near optimal Searching and online Recognition in Star Polygons," *Proc. 5th International Workshop on Algorithms and Data Structures - WADS* (1997), 284–296.

16. C. Papadimitriou and M. Yannakakis, "Shortest paths without a Map," *Theoretical Computer Science* 84 (1991), 127–150.

17. F.P. Preparata and M.I. Shamos, *Computational Geometry: an Introduction*, Springer-Verlag, 1985.

18. G. Rote, "Curves with increasing chords," *Mathematical Proceedings of the Cambridge Philosophical Society* 115 (1994), 1–12.

19. S. Schuierer and I. Semrau, "Eine optimale Suchstrategie für Straßen," *Technical Report 97*, Institut für Informatik, Universität Freiburg, Germany, 1998.

20. D. Sleator and R.E. Tarjan, "Amortized efficiency of list update and paging rules," *Communications of the ACM* 28 (1985), 202–208.

Department of Computer Science
University of Ioannina

Technical Reports

1998

1-98: G.D. Akrivis, V.A. Dougalis, O.A. Karakashian, W.R. McKinney
"Numerical Approximation of Singular Solutions of the Damped. Nonlinear Schrödinger Equation"

2-98: P.A. Voltairas, D.I. Fotiadis, C.V. Massalas
"Micromagnetics of Thin Ferromagnetic Films under Mechanical Stress"

3-98: A. Likas, V.Th. Paschos
"A new solution representation for the travelling salesman problem"

4-98: D.G. Papageorgiou, I.N. Demetropoulos and I.E. Lagaris
"MERLIN - 3.0: A Multidimensional Optimization Environment"

5-98: D.G. Papageorgiou, I.N. Demetropoulos and I.E. Lagaris
"The Merlin Control Language for Strategic Optimization"

6-98: E. Pitoura and I. Fudos
"Address Forwarding in Hierarchical Location Directories for Mobile PCS Users"

7-98: I.E. Lagaris, A. Likas, D.G. Papageorgiou
"Neural Network Methods for Boundary Value Problems Defined in Arbitrarily Shaped Domains"

8-98: D.I. Fotiadis, G. Foutsitzi and Ch.V. Massalas
"Wave Propagation in Human Long Bones"

9-98: G. Samaras, E. Pitoura
"Computational Models for Wireless and Mobile Environments"

10-98: E. Pitoura
"Transaction-Based Coordination of Software Agents"

11-98: A. Charalambopoulos, D.I. Fotiadis and C.V. Massalas
"Biomechanics of the Human Cranial System"

12-98: A. Charalambopoulos, D.I. Fotiadis and C.V. Massalas
"Dynamic Response of the Human Head to an External Stimulus"

13-98: E. Pitoura
"Supporting Read-Only Transaction in Wireless Broadcasting"

14-98: S.D. Nikolopoulos
"On the Structure of A-free Graphs"

15-98: A. Ktenas, D.I. Fotiadis and C.V. Massalas
"Comparison of Nucleation Models for Inhomogeneous Ferromagnetic Materials"

16-98: A. Likas
"Multivalued Parallel Recombinative Reinforcement Learning: A Multivalued Genetic Algorithm"

17-98: A. Likas, A. Stafylopatis
"Training the Random Neural Network Using Quasi-Newton Methods"

18-98: D.I. Fotiadis, G. Foutsitzi and C.V. Massalas
"Wave Propagation Modeling in Human Long Bones"

19-98: P.A. Voltairas, D.I. Fotiadis and C.V. Massalas
"Effect of Magnetostriction on the Magnetization Reversal of a Fine Particle in a Solid Non-Magnetic Matrix"

20-98: E. Pitoura and G. Samaras
"Locating Objects in Mobile Computing"

21-98: E. Pitoura
"Scalable Invalidation-Based Processing of Queries in Broadcast Push Delivery"

22-98: A. Likas
"Probability Density Estimation Using Multilayer Perceptrons"

23-98: K.N. Malizos, M.S. Siafakas, D.I. Fotiadis, P.N. Soucacos
"Semiautomated Volumetric Description of Osteonecroses"

24-98: N. Vlassis, A. Likas
"The Kurtosis-EM algorithm for Gaussian mixture modelling"

25-98: P. Rondogiannis
"Stratified Negation in Temporal Logic Programming and the Cycle Sum Test"

26-98: E. Pitoura, P. Chrysanthis
"Scalable Processing of Read-Only Transactions in Broadcast Push"

27-98: D. Barelos, E. Pitoura, G. Samaras
"Mobile Agent Procedures: Metacomputing in Java"

28-98: A. Likas
"A Reinforcement Learning Approach to On-line Clustering"

29-98: P.A. Voltairas, D.I. Fotiadis and C.V. Massalas
"The Role of Material Parameters and Mechanical Stresses on Magnetic and Magnetostrictive Hysteresis"

Department of Computer Science
University of Ioannina

Technical Reports

1999

1-99: V.V. Dimakopoulos
Single-Port Multinode Broadcasting

2-99: E. Pitoura, P.K. Chrysanthis
Exploiting Versions for Handling Updates in Broadcast Disks

3-99: P. Rondogianiis and M. Gergatsoulis
"The Branching-Time Transformation Technique for Chain Datalog Programs"

4-99: P.A. Voltairas, D.I. Fotiadis and C.V. Massalas
"Nonuniform Magnetization Reversal In Stressed Thin Ferromagnetic Films"

5-99: S.D. Nikolopoulos
"Coloring Permutation Graphs in Parallel"

6-99: A.P. Liavas
"Least-squares channel equalization performance versus equalization delay"

7-99: I. Fudos and L. Palios
"An Efficient Shape-based Approach to Image Retrieval"

8-99: D.I. Fotiadis, G. Foutsitzi and C.V. Massalas
"Wave Propagation in Human Long Bones of Arbitrary Cross Section"

9-99: V.S. Kodogiannis, E.M. Anagnostakis
"Soft Computing Based Techniques for Short-term Load Forecasting"

10-99: V.S. Kodogiannis
"A Study on Attitude Controle Schemes Based on Neural Network Structures"

11-99: V.S. Kodogiannis
"A Neural Network-Based Detection Thresholding Scheme for Active Sonar Signal Tracking"

12-99: V.S. Kodogiannis
"Compurison of Advanced Learning Algorithms for Short-term Load Forecasting"

13-99: V.S. Kodogiannis
"Development of a Breast Cancer Diagnostic System Via a Fuzzy-genetic Scheme"

14-99: P. Rondogiannis
"Adding Multidimensionality to Procedural Programming Languages"

15-99: P.A. Voltairas, D.I. Fotiadis and C.V. Massalas
"Magnetization Curling in an Elastic Sphere
(An pper Bound to the Nucleation Field)

16-99: A. Charalambopoulos, G. Dassios, D.I. Fotiadis and C.V. Massalas
"Scattering of a Point Generated Field by a Multilayered Spheroid"

17-99: A. Ktena, D.I. Fotiadis and C.V. Massalas
"A New 2-D Model for Inhomogeneous Permanent Magnets"

18-99: P.A. Voltairas, D.I. Fotiadis and C.V. Massalas
"Inverse Magnetostriction Effect in the Ferromagnetic Films
(A Micromagnetic Model)"

19-99: A. Likas and I.E. Lagaris
"The Moving Targets Training Algorithm for Classification Neural Netwoks"

20-99: E. Pitoura and I. Fudos
"Distributed Location Databases for Tracking Highly Mobile Objects"

21-99: A. Ktena, D.I. Fotiadis and C.V. Massalas
"A 2-D Preisach Model for Inhomogeneous Magnets"

22-99: A.P. Liavas
"Subspace methods for blind symbol estimation: The unknown channel order case"

23-99: P.A. Voltairas, D.I. Fotiadis, C.V. Massalas
"Mode Mixing in Ferromagnetic Resonance in Magnetic Microspheres"

24-99: A. Likas
"Reinforcement Learning Using the Stochastic Fuzzy Min-Max Neural Network"

25-99: M. Kano and S.D. Nikolopoulos
"On the Structure of A-free Graphs: Part II"

26-99: E. Pitoura, P.K. Chrysanthis, K. Ramamritham
"A Theory of Currency and Consistency in Broadcast Datavases"

27-99: P.A. Voltairas, D.I. Fotiadis and C.V. Massalas
"Estimation of exchange constant A and g factor for $Co_{\chi}Ni_{1-\chi}$ microspheres from size dependent ferromagnetic resonance modes"

28-99: G. Akrivis and Ch. Makridakis
"Convergence of a Time Discrete Galerkin Method for Semilinear Parabolic Equations"

29-99: S. NIKOLOPOULOS

30-99: S. NIKOLOPOULOS

31-99: G.D. Akrivis, V.A. Dougalis and G.E. Zouraris
"Finite Deference Schemes for the "Parabolic" Equation in a Variable Depth Environment with a Rigid Bottom Boundary Condition"

32-99: A. Charalambopoulos, D.I. Fotiadis and C.V. Massalas
"On the Solution of Boundary Value Problems using Spheroidal Eigenvectors"

33-99: L. Palios
"Reaching the Kernel of an Unknown Polygon Fast"

34-99: L. Palios
"Linear-time Algorithms for Tree Partitions with Small Cutsize"