On-line path planning in an unknown polygonal environment

Amitava Datta *, Subbiah Soundaralakshmi

School of Computer Science and Software Engineering, University of Western Australia, Perth, WA 6009, Australia

Received 30 December 2001; received in revised form 1 April 2003; accepted 8 May 2003

Abstract

We consider the navigation problem of a robot from a starting point $S$ to a target point $T$ inside an unknown rectilinear polygon. The robot knows the directions of the points $S$ and $T$ and it can detect the edges of the polygon through its tactile sensors. We present a competitive strategy for the robot to reach the target $T$. The competitive ratio of our strategy is $\frac{4.5d(\kappa - 1)(\kappa + 2) + O(1)}{\kappa}$, where, $\kappa = \frac{3(d/2)-1}{d}$ and $d$ is the $L_1$ distance between $S$ and $T$. We do not assume any structural restriction on the polygon, however the point $S$ must be on one of the extreme edges of the polygon. This is the first strategy to search for a target inside an arbitrary unknown polygon with competitive ratio independent of the number of edges of the polygon.

© 2003 Elsevier Inc. All rights reserved.

1. Introduction

Finding a path from a starting position to a target position is an important problem in robotics. Recently, this problem has been investigated extensively when the robot does not have any prior knowledge of its environment. The robot gathers local information through its on-board sensors and tries to find a “good” path from $\text{Start} (S)$ to $\text{Target} (T)$. Such search strategies are called on-line in the framework pioneered by Sleator and Tarjan [16]. In this
framework, if the path-length ($P$) generated by the robot is related to the shortest path ($ST$) from $S$ to $T$ by $P \leq c \times ST$, then the robot’s strategy is called $c$-competitive, and the ratio $c$ is called the competitive ratio achieved by the strategy.

Papadimitriou and Yannakakis [15] first investigated the problem of path planning in unknown environment in the framework of on-line algorithms. They proved that for axes-oriented square obstacles, the best achievable competitive ratio is $\frac{3}{2}$ and they gave a matching strategy which achieves this ratio. For axes-parallel rectangular obstacles, they proved that the best achievable competitive ratio is $\sqrt{n}$, when the point $T$ is at a horizontal distance of $n$ from $S$. Later, Blum et al. [4] gave a strategy with a matching competitive ratio of $\sqrt{n}$ for such an environment. Recently, Berman et al. [3] have beaten the deterministic lower bound in [15] by providing a randomized strategy with competitive ratio of $O(n^{1/3}\log n)$. In [3,4,15], the performance of the robot is almost unaffected (except for a constant factor in the competitive ratio) even when it uses a tactile sensor instead of a camera.

Klein [10] started the study of on-line algorithms for planning the path of a robot in a bounded region like a polygon. The robot has to find a path from $S$ to $T$ depending on the local information it gathers through its on-board cameras. He gave a competitive strategy for searching a restricted class of polygons called streets such that the two polygonal chains from $S$ to $T$ are mutually weakly visible. The competitive ratio for searching streets has been improved later by Kleinberg [11], Icking [9] and Lopez-Ortiz and Schuierer [12]. Datta and Icking [6] presented an optimal competitive strategy for searching a more general class of rectilinear polygons called generalized streets and later Datta et al. [5] presented a competitive strategy for searching a still more general class of rectilinear polygons called HV streets. Recently, Lopez-Ortiz and Schuierer [13] have presented a competitive strategy for searching arbitrary (non-rectilinear) generalized streets.

Even though the papers mentioned above explore interesting classes of polygons which allow constant-competitive searching, there is no known online algorithm for searching arbitrary unknown polygons with a bounded competitive ratio. The only previous work on searching arbitrary (without any structural restriction) unknown polygons is due to Kleinberg [11]. He showed that there is a strategy using vision-based sensors for searching arbitrary unknown polygons which achieves a competitive ratio of $O(m)$, where $m$ is the number of essential cuts in the polygon. But in the worst case $m = O(n)$ and as a result, the competitive ratio is $O(n)$. This upper bound matches the lower bound provided by Klein [10] where he constructed a simple polygon to show that no strategy can achieve a competitive ratio better than $\Omega(n)$ when the robot is equipped with a camera and can only recognize the target by seeing it. It is an interesting question whether it is possible to improve upon this competitive ratio in this model if the robot knows the direction of the target as well.
From the resource point of view, our model is different from the model considered in [5,6,10–13]. In these strategies, the robot is equipped with a camera and it does not have any knowledge about the location of the target $T$. We assume that the robot has only tactile sensors and it knows the directions of the points $S$ and $T$. The robot neither has a camera nor has any knowledge of its current coordinates at any point on its path. Our model is not unrealistic for designing mobile robots navigating in a workspace or factory floor. It is possible to install radio beacons at the points $S$ and $T$ so that the robot can track the exact directions of these two points from any point on its path. Such a model has been used by Lumelsky and Tiwari in [14] where they consider robot navigation with azimuth inputs. In our opinion, such angular or directional inputs improve upon two previous models extensively considered in the literature.

In case of a vision-based sensor like a camera, the robot spends considerable amount of time in processing the image of its surroundings. As a result, the path length generated by a robot under these strategies is not a good measure of the performance of the robot. Further, there may be errors in processing such images and the accumulated errors may affect the convergence of the strategies. Most of the papers which assume a vision-based sensor do not discuss error handling. In the other model, when the robot is assumed to know its current coordinates at every point of its path, accumulation of errors is again a serious problem. Usually, a robot calculates its current coordinates by integrating odometric information like total number of turns of its wheels. But due to friction and frequent turning, such odometric information is prone to error which is cumulative.

In our model, the robot only detects the directions of the points $S$ and $T$ and hence, we expect that this detection requires much less computation compared to image processing and there is no possibility of accumulation of errors. In our opinion, our model is simpler and not prone to errors compared to the models which either assume vision-based sensors or exact coordinates of the robot. However, such a conjecture is yet to be proved in a manner similar to Donald [8].

The only restriction in our model is that the point $S$ is on one of the extreme edges of the polygon. But the polygon can have arbitrary structure. In our strategy, the robot does only simple computations as opposed to the strategies in [5,6,10–13] where the robot spends considerable amount of time in image processing.

Our contribution in this paper is a competitive strategy for searching unknown and arbitrary rectilinear polygons with a competitive ratio $\leq 4.5d(\kappa - 1)(\kappa + 2) + O(1)$, where $\kappa = 3^{d/2} - 1$ and $d$ is the $L_1$ distance from $S$ to $T$. So, this competitive ratio is independent of $n$, the number of edges of the polygon. Even though the competitive ratio is quite high in the worst case, we expect that this ratio will be much smaller in most practical polygons. Also,
from a theoretical point of view we can say that only directional information enables us to achieve a competitive ratio independent of $n$ and hence, such input is more important compared to local visibility maps.

The rest of the paper is organized as follows. In Section 2, we present our model and in Section 3 we present and analyse our strategy.

2. The model

In this section, we state our model of the environment and discuss some fundamentals of our strategy.

Our model of the environment is the following.

• A point robot has to travel from a starting point $S$ to a target point $T$ inside a simple arbitrary rectilinear polygon. The points $S$ and $T$ are on the polygon boundary. Further, the point $S$ is on one of the extreme edges of the polygon (with respect to either $x$- or $y$-coordinate). In the rest of the paper, we assume without loss of generality that $S$ is on the least $x$-coordinate edge of the polygon and $T$ has both of its $x$- and $y$-coordinates greater than that of $S$. In the directional sense, we say that $T$ is towards the north-east of $S$. There are two radio beacons at the two points $S$ and $T$. The robot is equipped with a compass and a detector so that it can determine the absolute compass directions and the directions of $S$ and $T$ with respect to these compass directions. The robot also has tactile sensors and it can detect the polygon edges when it bumps onto them.

• We assume that the complete geometry of the environment is based on the notion of a certain “unit” length. Each side of the rectilinear polygon is at least of this length and the $x$ or $y$ distance between two parallel sides is also at least of this length. These assumptions are realistic since no real-life environment has sides or gaps of infinitesimally small length. Whenever we talk about $d$, the $L_1$ distance between $S$ and $T$, we assume that the distance $d$ is measured in terms of this unit length.

Our model is partially similar to the models in [3,4,15] as these models assume tactile sensors for detecting the edges of the obstacles. However, these papers assume that the robot always knows its current absolute position and the position of the target. In our model, the robot only knows the directions of the points $S$ and $T$. In this sense, our model is similar to that considered by Lumelsky and Tiwari [14], where they consider target reachability with azimuth inputs. However, they do not analyse their strategy in the framework of competitive analysis. Our model is different from the model recently considered by Angluin et al. [1] where they have considered robot navigation with range queries.
We use an on-line strategy due to Baeza-Yates et al. [2] for searching for a target in two concurrent rays. A point robot is initially placed at the origin of the two rays and it does not know the position of the target. The target lies on one of the rays. It turns out that [2] the optimal competitive ratio the robot can achieve is through a doubling technique and this ratio is 9. The robot traverses the two rays alternately and every time it doubles the distance it walks. The total distance walked by the robot is \(9l\), where the distance from the origin of the two rays to the target point is \(l\) units. Suppose, \(\rho_1\) and \(\rho_2\) are the two concurrent rays and the target is at a distance of \(l\) units from the origin on \(\rho_1\).

**Lemma 2.1.** If a point robot follows the strategy in [2], the furthest distance walked by the robot along \(\rho_2\) is \(2(l - 1)\) units.

**Proof.** In the worst case, the robot will turn back from \(\rho_1\) when it is at a distance 1 from the target, i.e., at a distance \((l - 1)\) from the origin. It will then travel a distance \(2(l - 1)\) along \(\rho_2\) and the next time it goes along \(\rho_1\), it will reach the target (Fig. 1). \(\square\)

In our strategy, the robot does not execute this doubling strategy along two concurrent rays, rather it goes along the edges of the polygon. The step-length and distance walked by the robot is measured along the edges and from the starting point in terms of the unit length we have discussed before.

### 3. The strategy for searching an unknown rectilinear polygon

We indicate the coordinates of the points \(S\) and \(T\) as \((x_S, y_S)\) and \((x_T, y_T)\) respectively. We assume without loss of generality that \(x_S \leq x_T\) and \(y_S \leq y_T\), i.e., \(T\) is to the right and above \(S\). Our strategy remains valid and can be easily modified for the other cases.

For a point \(p_i\), we represent its \(x\)- and \(y\)-coordinates by \(x_i\) and \(y_i\) respectively. We refer to Fig. 2 for illustration of the following definitions.

**Vertical and horizontal target lines:** Consider the part of the line \(x = x_T\) starting at \((x_T, y_T)\) and until the intersection point of \(x = x_T\) and \(y = y_S\). We call...
this line as the *vertical target line*. Similarly, the part of the line \( y = y_T \) starting at \((x_T, y_T)\) and until the intersection point of \( x = x_S \) and \( y = y_T \) is called the *horizontal target line*.

**Vertical and horizontal source lines:** The horizontal and vertical source lines are defined in a similar way. The part of the line \( y = y_S \) starting at \((x_S, y_S)\) and until the intersection point of \( x = x_T \) and \( y = y_S \) is called the *horizontal source line*. The part of the line \( x = x_S \) starting at \((x_S, y_S)\) and until the intersection point of \( x = x_T \) and \( y = y_T \) is called the *vertical source line*.

The following definitions are given for the vertical target line and the horizontal source line. These definitions for the horizontal target line and vertical source line are similar.

- Suppose the horizontal source line intersects the polygon boundary at \( x \)-coordinates \( x_1, x_2, \ldots, x_n \), such that \( x_1 < x_2 < \cdots < x_n \). We call the points \( p_1 = (x_1, y_S), p_2 = (x_2, y_S), \ldots, p_n = (x_n, y_S) \) as the *source event points*. Similarly, suppose the vertical target line intersects the polygon boundary at \( y \)-coordinates \( y_1, y_2, \ldots, y_m \), such that \( y_1 < y_2 < \cdots < y_m \). We call the points \( q_1 = (x_T, y_1), q_2 = (x_T, y_2), \ldots, q_m = (x_T, y_m) \) as the *target event points*.
- If the intersection of the lines \( x = x_T \) and \( y = y_S \) is inside the polygon, we call this intersection point \((x_T, y_S)\) as the 0th target event point.
- Consider a target event point \( q_i = (x_T, y_i) \). If the target event line towards the higher \( y \)-coordinate side of \( q_i \) lies inside (resp. outside) the polygon \( P \), \( q_i \) is called a *free* (resp. *blocked*) target event point. Note that, the free and blocked target event points appear alternately in the list of target event points.
• If a target event point is within the $y$-interval $[y_T, y_S]$, it is called an internal target event point. If a target event point has $y$-coordinate greater than $y_T$, it is called an external target event point. In the rest of the paper, by a target event point, we will always mean an internal target event point.

• The free and blocked source event points are defined in a similar fashion. For a source event point $q_i = (x_i, y_S)$, if the source event line towards the higher $x$-coordinate side of $q_i$ lies inside (resp. outside) the polygon, $q_i$ is called a free (resp. blocked) source event point. In the rest of the paper, by a source event point, we always mean an internal source event point, i.e., a source event point between $S$ and the target line.

In our strategy, the robot tries to go from an event point to the next event point. Note that, for two consecutive target event points $q_k = (x_T, y_k)$ and $q_l = (x_T, y_l)$ with $y_k < y_l$, if $q_k$ is a free target event point and $q_l$ is a blocked target event point, the vertical line segment joining $q_k$ and $q_l$ lies completely within the polygon $P$. We call such a segment of the target line joining a free target event point to the next blocked target event point as a free target segment. A free source segment is defined in a similar way.

In Fig. 2, $p_1$ and $p'_1$ are respectively examples of blocked and free source event points on the horizontal source line. Similarly, $q_1$ and $q'_1$ are examples of blocked and free target event points on the vertical target line. The horizontal segment starting at $p'_1$ and until it hits the polygon boundary is a free source segment. Similarly, the vertical segment starting at $q'_1$ and until it hits the polygon boundary is an example of a free target segment. A shortest path from $S$ to $T$ is shown by the bold dashed line.

We now discuss some sensing capabilities of the robot in the light of the above definitions.

**Lemma 3.1.** At any point of its path, the robot can decide whether it is on one of the source or target lines.

**Proof.** We only consider how the robot determines whether it is on the horizontal source line. The other cases are similar. Note that, the robot can determine the absolute compass directions with its compass. If the direction of the radio beacon at the point $S$ is directly towards east of the present position of the robot, the robot is on the horizontal source line. □

The robot can also extract distance information through its sensors in the following way.

**Lemma 3.2.** Given two points $\tau_1$ and $\tau_2$ on the vertical (resp. horizontal) target line, the robot can determine which one of them is closer to the target.
**Proof.** We only consider the case when the two points are on the vertical target line (refer Fig. 3). The other case is similar. The robot can determine the angle \( \angle S \tau_1 T \) between the two lines \( \overline{T \tau_1} \) and \( \overline{T \tau_1} \). Suppose this angle is \( \alpha \). Similarly, suppose the angle between the two lines \( \overline{T \tau_2} \) and \( \overline{T \tau_2} \) is \( \beta \). If \( \alpha < \beta \) (resp. \( \alpha > \beta \)), \( \tau_2 \) (resp. \( \tau_1 \)) is closer to the point \( T \).

**Corollary 3.3.** Given two points \( \tau_1 \) and \( \tau_2 \) on the vertical (resp. horizontal) source line, the robot can determine which one of the points is closer to the horizontal (resp. vertical) target line.

### 3.1. The strategy

There are two phases in our strategy. In the first phase, starting from \( S \), the robot reaches one of the target lines. In the second phase, starting from the first point on one of the target lines, the robot reaches the target \( T \). The robot applies similar strategies in both phases and we describe below only the first phase. Essentially the robot takes two actions as described below. It has two registers \( R_1 \) and \( R_2 \). For the horizontal (resp. vertical) source line, the robot stores the angle \( \angle T p_i S \) of the blocked event point \( p_i \) closest to the vertical (resp. horizontal) target line in the register \( R_1 \) (resp. \( R_2 \)). Both \( R_1 \) and \( R_2 \) are initialized to \( \infty \). Note that these two angles decrease as the robot reaches blocked event points progressively closer to the two target lines.

**Case 1 (motion in free space):** If the robot is at a free horizontal (resp. vertical) source event point \( p_k \) and \( \angle T p_k S < R_2 \) (resp. \( \angle T p_k S < R_1 \)), it walks along the free source segment starting at \( p_k \). Initially, if \( S \) is a free source event point on the horizontal (resp. vertical) source line, the robot walks along the
free horizontal (resp. vertical) source segment starting at \( S \). If \( \angle T_p S > R_2 \) (resp. \( \angle T_p S > R_1 \)), the robot continues walking along the polygon boundary.

Case 2 (motion along the polygon boundary): If the robot is at a blocked horizontal (resp. vertical) source event point \( p_m \) and \( \angle T_{p_m} S < R_1 \) (resp. \( \angle T_{p_m} S < R_2 \)) it takes two actions. First, it stores \( \angle T_{p_m} S \) in \( R_1 \) (resp. \( R_2 \)) and then it doubles along the polygon boundary with \( p_m \) as the origin according to the strategy in [2]. If \( \angle T_{p_m} S > R_1 \) (resp. \( \angle T_{p_m} S > R_2 \)), the robot continues walking along the polygon boundary.

Intuitively, the robot’s strategy is the following. It walks along a source segment if the corresponding free source event point on the vertical (resp. horizontal) source line is closer to the horizontal (resp. vertical) target line compared to all the previous vertical (resp. horizontal) source event points it has reached. Similarly, it doubles along the polygon boundary starting at a blocked source event point if this point is closer to one of the target lines compared to all the previous source event points it has reached.

The strategy in the second phase is almost the same. In this phase, the points \( p_k \) and \( p_m \) indicate free and blocked target event points respectively. And the robot stores in \( R_1 \) (resp. \( R_2 \)) the angle of the blocked target event point which it has encountered closest to the target on the horizontal (resp. vertical) target line. The robot can decide that it has reached \( T \) by detecting a discontinuity in the direction of \( T \) after it passes through \( T \). From now on, we will refer to the robot’s strategy as algorithm Polygon_Search.

In Fig. 4, the robot’s directions of doubling along the polygon boundary are shown by dotted arrows. The directions of travel along the free segments are shown by dashed arrows.
3.2. Analysis of the strategy

We represent the shortest $L_1$ path from $S$ to $T$ as $SP$. It is possible that there may be more than one shortest path, but we consider only one such shortest path.

**Definition 3.4.** Suppose, the maximum (resp. minimum) $x$-coordinate that $SP$ reaches is $x_{\text{max}}$ (resp. $x_{\text{min}}$). We call the $x$-interval $[x_{\text{max}}, x_{\text{min}}]$ as the critical $x$ length or $\mathcal{X}$ in short. Similarly, if the maximum (resp. minimum) $y$-coordinate that $SP$ reaches is $y_{\text{max}}$ (resp. $y_{\text{min}}$), we call the $y$-interval $[y_{\text{max}}, y_{\text{min}}]$ as the critical $y$ length or $\mathcal{Y}$ in short. Finally, the rectangular area formed by taking the product of these two intervals $\mathcal{X}$ and $\mathcal{Y}$ i.e., $[x_{\text{max}}, x_{\text{min}}] \times [y_{\text{max}}, y_{\text{min}}]$ is called the critical box (Fig. 5).

In our analysis, we assume an adversary who knows the strategy of the robot and can design the polygon in such a way that the robot is forced to walk as much distance as possible following the strategy. In the following analysis, we construct a polygon such that the adversary forces the robot to walk a maximum distance following our strategy compared to a shortest path from $S$ to $T$.

We first state the convergence of our strategy.

**Lemma 3.5.** The robot correctly reaches the target by executing Algorithm Polygon_Search.

**Proof.** In the simplest case, the robot does not encounter any source or target event points except the 0th target event point. In this case, the robot walks directly to the 0th target event point and then to the target $T$.

In general, the robot will encounter at least one source event point $p_i$ and starting from $S$, $p_i$ must be a blocked source event point (even the point $S$ itself may be this blocked event point). The robot executes Case 2 (motion along the polygon boundary), i.e., starts doubling along the polygon boundary with $p_i$ as
the origin. Suppose, there are other source or target event points on the polygon boundary. Since the polygon boundary is a closed curve, the robot will eventually reach either a source event point \( p_j \) such that \( x_j > x_i \), or the target line. We assume that the robot reaches a source event point like \( p_j \) first. There are two possibilities, \( p_j \) may be either a blocked or a free event point.

- If \( p_j \) is a blocked event point, it starts doubling with \( p_j \) as the origin and eventually reaches a source event point \( p_k \) such that \( x_k > x_j \) (refer Fig. 4).
- If \( p_j \) is a free event point, it travels along the free segment starting at \( p_j \) and reaches a blocked event point closer to the target line compared to \( p_j \) (refer Fig. 4).

Following this scheme, it will progressively reach source event points closer to the target line and will eventually reach the target line.

Following the same argument, starting from a target event point, the robot progressively reaches other target event points closer to the target and ultimately the target \( T \). Note that the robot never goes into a cycle since it travels along the free segments only when a free event point is closer to the target line or to \( T \) compared to the previous blocked event point. Since \( T \) is on the polygon boundary, the robot will eventually reach \( T \) either along a free target segment or along one of the search paths while doubling at a blocked event point.

We now analyse the competitive ratio of our strategy. First, we explain some geometric properties which will be useful in this analysis.

**Lemma 3.6.** The maximum number of blocked and free source event point pairs on the horizontal source and target lines is \( \mathcal{X}/2 \) each. And the maximum number of blocked and free target event point pairs on the vertical source and target lines is \( \mathcal{Y}/2 \) each.

**Proof.** In our model, every edge as well as the gap between two parallel edges is at least of unit length. Since the length of the source line is at most \( \mathcal{X} \), the total number of blocked and free source event point pairs can be at most \( \mathcal{X}/2 \). The other bound can be proved in a similar way.

**Lemma 3.7.** The maximum possible perimeter of a rectilinear polygon inscribed in an \( m \times n \) grid is \( mn \).

**Proof.** The maximum perimeter polygon should pass through every grid point. Since a polygon is a non-intersecting curve, every grid point can have at most two polygon edges incident on it. Every edge is incident on two grid points. Since the total number of grid points is \( mn \), the maximum possible perimeter is \( mn \times 2/2 = mn \) (Fig. 6).
Corollary 3.8. The maximum length of the polygon boundary within the critical box is $XY$.

From Corollary 3.8, it is clear that the adversary will try to force the robot to walk outside the critical box as much as possible, since any distance walked by the robot inside the critical box is bounded by $XY$. However, in our strategy, the robot always maintains one search path inside the critical box so that the total path-length of the robot can be bound by some function of $d$. We discuss the adversary’s approach in two phases. In the first phase, we discuss how the adversary will construct the polygon such that the robot walks a maximum distance in going from $S$ to the first point on one of the target lines. In the second phase, we discuss the adversary’s construction from the first point on the target line the robot reaches and until it reaches $T$.

Lemma 3.9. The adversary will design the polygon in a way such that the following two conditions are satisfied. (i) The robot reaches the maximum number of source and target event points in the scene, and (ii) there is a shortest path which does not have a common point with the source and target lines except at the points $S$ and $T$.

Proof. From Lemma 3.6, it is clear that the total lengths of all the source (resp. target) free segments are bounded by $X/2$ (resp. $Y/2$). Hence, to make the robot walk a maximum distance, the adversary will try to make it walk the maximum distance along the polygon boundary between a blocked (source or target) event point to the next free (source or target) event point. This can be achieved by forcing the robot to reach all the event points. This proves claim (i). To prove claim (ii), note that, if the shortest path has some common points with the source and target lines, part of the robot’s path is consumed in the shortest path and as a result, the length of the robot’s path cannot be maximized. □
We next prove that the adversary has to further refine its construction of the polygon to make the robot walk a maximum distance before it reaches the target line. We assume that the robot initially walks along the horizontal source line. Consider a blocked source event point $p_k$ on the horizontal source line (refer Fig. 7). When the robot executes doubling with $p_k$ as the origin, there are two paths along the polygon boundary $\pi_j$ and $\pi'_j$. In the following discussion, by a source event point, we mean a source event point on the horizontal source line unless mentioned otherwise.

**Lemma 3.10.** At least one of the two paths $\pi_j$ and $\pi'_j$ lies inside the critical box until one of the following conditions occurs: (i) it reaches a source event point $p_m$ such that $x_m > x_k$, or (ii) it reaches a source event point $p_i$ such that $x_i < x_k$, or (iii) it reaches the target line.

**Proof.** First, we prove that both the paths $\pi_j$ and $\pi'_j$ cannot go out of the critical box without at least one of them satisfying one of the three conditions in the statement of the lemma.

Suppose, both the paths go out of the critical box and $p_j$ and $p'_j$ are two points on $\pi_j$ and $\pi'_j$ which the robot reaches while doubling (Fig. 7). Since both $\pi_j$ and $\pi'_j$ are connected parts of the polygon boundary, the shortest path cannot intersect either of them. Hence, the shortest path must reach $T$ by going around either $p_j$ or $p'_j$. This implies that at least one of the points $p_j$ or $p'_j$ lies inside the critical box.

In the following discussion, we assume without loss of generality that the path $\pi_j$ remains inside the critical box initially. We now examine the ways in which $\pi_j$ may go out of the critical box. If $\pi_j$ goes out of the critical box, it has to cross either of the four bounding sides of the critical box (Fig. 8(i) and (ii)). The four bounding sides are represented by $l_i$, $i = 1, \ldots, 4$. Parts of the shortest
path are shown by the bold dashed lines and parts of the polygon boundary are shown by solid lines. Note that the path $\pi_j$ cannot cross the line $l_1$ as the point $S$ is on the least $x$-coordinate edge of the polygon.

In the first case, suppose, the shortest path reaches $T$ by going above and around the path $\pi_j$ (Fig. 8(i)). In this case, since $\pi_j$ goes along the polygon boundary, it can have at most an overlap with the shortest path or the bounding line $l_2$. $\pi_j$ cannot cross $l_2$ since the shortest path goes around $\pi_j$. If $\pi_j$ crosses the line $l_3$, it reaches a target line (condition (iii)). And if it crosses $l_4$, it reaches a source event point (conditions (i) and (ii)).

In the second case, suppose the path $\pi_j$ crosses the bounding line $l_2$ of the critical box (Fig. 8(ii)). In this case, the shortest path must go around the path $\pi'_j$ to reach $T$. In other words, $\pi'_j$ remains completely within the critical box and $\pi'_j$ will eventually reach some point on the source or target line satisfying the conditions of the lemma. □

**Remark.** From Lemma 3.10, we can infer that the adversary will try to avoid case (i). Since, if condition (i) is satisfied, the robot’s path from $p_k$ to $p_m$ is completely within the critical box. The adversary will try to avoid condition (iii) as long as possible since once the robot reaches the target line, its path length from $S$ to the target line cannot be increased. This leaves us with condition (ii). Suppose, $p_j$ is a blocked source event point and the two paths along which the robot executes doubling along the polygon boundary are $\pi_j$ and $\pi'_j$. Since the polygon is a closed curve, one of the paths $\pi_j$ or $\pi'_j$ will reach a source event point closer to the target line compared to $p_j$. Assume w.l.o.g. that the robot reaches a free source event point $p'_{j+1}$ such that $x_{j+1} > x_j$ while walking along $\pi'_j$. From the remark above, the adversary will design the polygon in such a way

![Fig. 8. (i) The shortest path reaches $T$ by going above and around the path $\pi_j$. (ii) The path $\pi_j$ crosses the bounding line $l_2$ of the critical box.](image-url)
that the robot reaches a source event point $p_i$ along $\pi_j$ instead of reaching the target line (Fig. 9). Also, from Lemma 3.10, the robot remains inside the critical box until it reaches $p_i$.

**Corollary 3.11.** Suppose, $p_1$ is the first blocked source event point that the robot reaches and $\pi_1$ is the path (starting at $p_1$) along the polygon boundary which remains completely inside the critical box and $\pi_1$ does not reach a source event point closer to the target line compared to $p_1$. In this case, $\pi_1$ will reach the target line before going out of the critical box.

**Lemma 3.12.** After crossing $p_i$ and following $\pi_j$ along the polygon boundary, the robot will reach a free event point $p_l$ such that $x_S < x_l < x_i$ when $\pi_j$ next intersects the source line.

**Proof.** We prove this by contradiction. In the first case, suppose $\pi_j$ reaches a source event point $p_k$ such that $x_k > x_j$ (Fig. 9(i)). Since the polygon boundary from $p_j$ to $p_k$ via $p_i$ is a connected curve, the robot could not have reached $p_{j-1}$ earlier except going through some point on the source line which is between $p_j$ and $p_k$. But in our strategy, the robot does doubling at a blocked source event point only if it has a greater $x$-coordinate compared to any previous source event point the robot has encountered. Hence, the robot would not have executed doubling at $p_j$ if it had reached a point with greater $x$-coordinate than $p_j$ before reaching $p_j$.

In the second case, suppose the robot reaches $S$ starting from $p_i$ along $\pi_j$ (refer Fig. 9(ii)). Since $p_j$ and $p_i$ are two consecutive intersections of the polygon boundary with the source line and the interior of the polygon is to the left of $p_j$ (since it is a blocked event point), the interior of the polygon should be towards the right of $p_i$. For the same reason, if $p_j$ and $S$ are two consecutive intersection points of the polygon boundary with the source line, the interior of the polygon should be to the left of $S$. But this is a contradiction since $S$ is on

![Fig. 9. (i) The path $\pi_j$ reaches source event point $p_k$ such that $x_k > x_j$. (ii) Robot reaches $S$ starting from $p_i$ along the path $\pi_j$.](image-url)
the least $x$-coordinate edge of the polygon. Hence, $\pi_j$ must have another intersection with the source line before $S$. \hfill $\Box$

**Lemma 3.13.** The robot walks a maximum distance in reaching the target line starting from $S$, if for every pair of blocked and the next free (source) event points $p_j$ and $p_j'$, the path $\pi_j'$ connecting $p_j$ and $p_j'$ along the polygon boundary lies completely outside the critical box. Moreover, the other path $\pi_j$ from $p_j$ should pass through all the previous source event points before it reaches the target line.

**Proof.** Suppose the two paths along the polygon boundary from $p_j$ are $\pi_j$ and $\pi_j'$ and the path $\pi_j'$ from $p_j$ to $p_j'$ lies outside the critical box (Fig. 10). From Lemma 3.10, the adversary will try to design the polygon in such a way that the other path $\pi_j$ does not reach either the target line or a source event point closer to the target line compared to $p_j$. As a result, $\pi_j$ must eventually reach a source event point $p_i$ closer to $S$ compared to $p_j$ (i.e., $x_i < x_j$) when it goes out of the critical box. From Lemma 3.12, after crossing $p_i$, the path $\pi_j$ must reach a source event point $p_l$ such that $x_S < x_l < x_j$. From Corollary 3.11, one of the paths from the first blocked event point reaches the target line. Note that, Lemma 3.12 is true for every blocked event point $p_j$.

Suppose the pairs of blocked and free event points along the source line are $(p_1, p'_1), (p_2, p'_2), \ldots, (p_j, p'_j)$ before a blocked source event point $p_j$, i.e., $x_S < x_1 < x'_1 < x_2 < \cdots x'_{j-1} < x_j$ (refer Fig. 11). From Lemma 3.12, one of the paths from $p_2$ will pass through $p'_1$ and $p_1$. Similarly, one of the paths from $p_3$ will at least pass through one of the pairs $(p_2, p'_2)$ or $(p_1, p'_1)$. Note that, if the path from $p_3$ passes through $(p_2, p'_2)$, it will pass through $(p_1, p'_1)$ as well. From Corollary 3.11, such a path will eventually reach the target line. Hence, the adversary’s best strategy is to let the robot walk as much distance as possible before it reaches the target line. As a result, for a blocked event point $p_j$, the adversary will design the polygon in such a way that the robot walks through all the previous source event point pairs before it reaches the target line. Then

Fig. 10. The path $\pi_j'$ from $p_j$ to $p_j'$ lies outside the critical box.
the adversary can make the length of the polygon boundary between $p_j$ and $p_j'$ twice the distance from $p_j$ to the target line (From Lemma 2.1). The adversary’s construction of the polygon is shown in Fig. 4. It consists of rakes of increasing lengths starting from the rake which has the first blocked event point on it.

**Remark.** It does not help to increase the path length of the robot if the adversary introduces a similar rake structure along the vertical source line. Such a rake structure will introduce two possibilities. In the first case, the path from a blocked event point to the next free event point (on the vertical source line) is within the critical box and the length of the robot’s path will be a constant multiple of the shortest path (actually nine times the length of this part of the shortest path from the result in [2]). In the second case, the shortest path will overlap the robot’s path between a blocked and the next source event point. In this case also, the length of the robot’s path will be a constant multiple of the shortest path (again nine times the length of this part of the shortest path). In both the cases, a constant factor will be introduced in the overall competitive ratio. As a result, the adversary cannot increase the robot’s path asymptotically if he/she introduces a rake structure along the vertical source line as well. In other words, the adversary can maximize the length of the robot’s path as a function of $d$ by adding a rake structure only along one of the two source lines, so that the shortest path does not have any overlap with the robot’s path.

We now estimate the length of the robot’s path in reaching the target line starting from $S$. We assume w.l.o.g that the first target event point the robot reaches is on the vertical target line (Fig. 12). Suppose, the first blocked (source) event point the robot reaches is $p_1$ and the first free (source) event point is $p_1'$. 

---

**Fig. 11.** $p_1, p_2, p_3, \ldots, p_j$ are blocked event points and $p_1', p_2', p_3', \ldots, p_{j-1}'$ are free event points on the source line.
Lemma 3.14. The adversary can place $p'_1$ at a distance $\leq 2x_1$ from $p_1$ such that the robot reaches $p'_1$ before it reaches the target line. Moreover, the robot traverses a distance $\leq 9x_1$ along the polygon boundary before it reaches $p'_1$ from $p_1$. Here, $x_1$ is a part of the polygon boundary completely inside the critical box.

Proof. From Lemmas 3.10 and 3.13, the path along the polygon boundary between $p_1$ and $p'_1$ is outside the critical box. Also, the other path starting at $p_1$ is inside the critical box until it reaches the target line. We call the path which is inside (resp. outside) the critical box as $\pi_1$ (resp. $\pi_2$). Suppose, the length of $\pi_1$ before it reaches the target line is $x_1$. Notice that, the target line may not be at a distance which is an integer multiple of the unit length along $\pi_1$. Hence, the robot may turn back from $p_1$ when it is an infinitesimal distance away from the target line, instead of one step away as in Lemma 2.1. As a result, the distance of $p'_1$ from $p_1$ can be $\leq 2x_1$. The total distance walked by the robot in reaching $p'_1$ is $\leq 9x_1$ from the result in [2].

We consider the $j$th pair of blocked and free source event points $p_j$ and $p'_j$ and estimate the distance between $p_j$ and $p'_j$ (we call it $\lambda_j$) along the polygon boundary.

Lemma 3.15. $\lambda_j \leq 2x_j + 2 \times 3x_{j-1} + 2 \times 3^2x_{j-1} + \cdots + 2 \times 3^{j-1}x_1$, where $x_k, k = 1, \ldots, j$ are parts of the polygon boundary completely inside the critical box.

Proof. From Lemmas 3.10 and 3.13, all the paths between a blocked event point and the next free event point are outside the critical box. $x_k, k = 1, \ldots, j$ are the parts of the robot’s path which are inside the critical box. We prove the...
Lemma 3.16. $\lambda_1 \leq 2x_1$. Similarly, $\lambda_2 \leq 2(x_2 + \lambda_1 + x_1) \leq 2x_2 + 2 \times 3x_1$. We assume the claim holds for $\lambda_{j-1}$, in other words, $\lambda_{j-1} \leq 2x_{j-2} + 2 \times 3x_{j-2} - 2 + 2 \times 3^{j-2}x_1$.

$\lambda_{j-1}$ is the distance between the $(j-1)$th pair of blocked and free event points, i.e., the distance between $p_{j-1}$ and $p'_{j-1}$. Now, from Lemma 3.14, the distance from $p_{j-1}$ to the target line is $\frac{1}{2}\lambda_{j-1}$. Since $\lambda_j$ is less than or equal to twice the distance of the target line from $p_j$, $\lambda_j \leq 2(\frac{1}{2}\lambda_{j-1} + x_j) = 2x_j + 3\lambda_{j-1} \leq 2x_j + 2 \times 3x_{j-1} + 2 \times 3^2x_{j-2} + \cdots + 2 \times 3^{j-2}x_1$. □

Suppose, the distance walked by the robot starting from $p_j$ until it reaches $p'_{j}$ is $\tau_j$.

**Lemma 3.16.** $\tau_j \leq 9 \times (x_j + 3x_{j-1} + 3^2x_{j-2} + \cdots + 3^{j-1}x_1)$.

**Proof.** We prove the lemma by induction. From Lemma 3.14, $\tau_1 \leq 9x_1$. Notice that, while doubling with the second blocked event point $p_2$, the distance of the target line from $p_2$ is $(x_2 + \lambda_1 + x_1)$. Hence, $\tau_2 \leq 9(x_2 + \lambda_1 + x_1)$ which simplifies to $\leq 9(x_2 + 3x_1)$. We assume that the lemma holds for $\tau_{j-1}$, i.e., $\tau_{j-1} \leq 9(x_{j-1} + 3x_{j-2} + 3^2x_{j-3} + \cdots + 3^{j-2}x_1)$.

Notice that, $\tau_j \leq \tau_{j-1} + 9(\lambda_{j-1} + x_j)$, where $\lambda_{j-1} \leq 2x_{j-1} + 2 \times 3x_{j-2} + 2 \times 3^2x_{j-3} + \cdots + 3^{j-2}x_1$. Hence,

$$
\begin{align*}
\tau_j & \leq \tau_{j-1} + 9(\lambda_{j-1} + x_j) \\
& \leq 9(x_{j-1} + (2 + 1)x_{j-1} + (2 + 1)3x_{j-2} + \cdots + (2 + 1) \times 3^{j-2}x_1) \\
& = 9(x_j + 3x_{j-1} + 3^2x_{j-2} + \cdots + 3^{j-1}x_1) \quad \square
\end{align*}
$$

**Lemma 3.17.** Starting from $S$, the total distance walked by the robot before it reaches the target line is: $\leq 4.5d^2(3^{(d/2)-1} - 1) + \frac{d}{2}$.

**Proof.** There are two components in the total distance walked by the robot, the distance along the polygon boundary and the horizontal parts along the free segments. The distance walked by the robot along the polygon boundary is $\sum_{j=1}^i \tau_j$, where $i$ is the number of blocked and free (source) event point pairs.

$$
\begin{align*}
\sum_{j=1}^i \tau_j & \leq 9 \left[ x_j + x_{j-1} \sum_{k=0}^{j} 3^k + x_{j-2} \sum_{k=0}^{j} 3^k + \cdots + x_1 \sum_{k=0}^{j-1} 3^k \right] \\
& \leq 9 \times \frac{3^{i-1} - 1}{3 - 1} \sum_{j=1}^i x_j \leq 4.5d^2(3^{(d/2)-1} - 1)
\end{align*}
$$

Notice that, from Lemma 3.6, $i \leq d/2 \leq d/2$. Also, $\sum_{j=1}^i x_j \leq 3d^2$. Since there can be a maximum of $d/2$ pairs of blocked and free event point pairs, the total distance walked by the robot along the horizontal free segments is
\[ X/2 \leq d/2. \] Hence the total distance walked by the robot to reach the target line is \[ \leq 4.5d^2 \left(3^{(d/2)^{-1}} - 1\right) + \frac{d}{2}. \]

Now, we estimate the distance walked by the robot before it reaches \( T \) starting from the first point on the target line. In the following discussion, by a target event point, we will mean a target event point on the vertical target line unless we specify otherwise. From Lemma 3.9, the adversary’s strategy would be to force the robot reach as many target event points as possible before it reaches \( T \). Since the target \( T \) is on the polygon boundary, either \( T \) itself is a free target event point or there is a free target event point \( p_k \) directly below \( T \) (i.e., the robot can reach \( T \) from \( p_k \) by walking along the free segment joining \( p_k \) and \( T \)). We call the portion of the boundary of the polygon walked by the robot while reaching the target line from \( S \) as \( \sigma \).

Suppose, the first blocked target event point the robot reaches is \( q_1 \). We indicate the two paths along the polygon boundary along which the robot executes doubling as \( P_1 \) and \( P_2 \).

**Lemma 3.18.** One of \( P_1 \) or \( P_2 \) reaches the target \( T \) or a point like \( p_k \) directly below \( T \) and includes \( \sigma \). The other path reaches \( T \) or a point like \( p_k \) and does not include \( \sigma \).

**Proof.** The proof follows from the fact that the polygon is a singly connected closed curve.

Suppose, the final source event point the robot reaches is \( p_i \). Since \( q_1 \) is a blocked target event point, one of the search paths \( P_1 \) or \( P_2 \) initially lies within the critical box.

**Lemma 3.19.** The robot walks a maximum distance in reaching \( T \) if for every pair of blocked and free target event points \( q_j \) and \( q_j' \), the path between \( q_j \) and \( q_j' \) is outside the critical box. Moreover, the path from \( q_j \) which does not reach \( q_j' \) should pass through all previous target event points and include \( \sigma \) as well before reaching \( T \).

**Proof.** We can prove in a manner similar to Lemma 3.13 that the robot walks a maximum distance if for a pair of target event points \( q_j \) (blocked) and \( q_j' \) (free), the polygon boundary between \( q_j \) and \( q_j' \) is outside the critical box. Also, for a blocked event point like \( q_j \), one of the paths along the polygon boundary from \( q_j \) passes through all previous target event points. It remains to be shown that such a path may also contain \( \sigma \). There are two possible ways in which the robot reaches the first blocked target event point \( q_1 \). It may reach \( q_1 \) either along a free segment or along the polygon boundary (Fig. 13(i)). If it reaches \( q_1 \) along a free segment, it must have reached the 0th target event point \( q_0 \) from the final
source event point (Fig. 13(i)). Note that the path $\Gamma$ along the polygon boundary starting at $q_1$ and to the left of the target line initially lies within the critical box. Since $\Gamma$ initially lies within the critical box, it should cross some source event point before going out of the critical box and the robot walks the maximum distance if the first source event point along $\Gamma$ is $p_i$. From Lemma 3.13, this path will include $\sigma$ after crossing $p_i$. If the robot reaches $q_1$ along the polygon boundary (Fig. 13(ii)), it has reached $q_1$ from $p_i$ and there is a path from $q_1$ to $p_i$. Since $p_i$ is a blocked source event point, the path $\Gamma$ from $p_i$ will eventually pass through all the previous source event points due to Lemma 3.13 and hence it includes $\sigma$. □

We assume that $\mathcal{P}_1$ includes $\sigma$ and $\mathcal{P}_2$ does not include $\sigma$ before reaching $T$. Hence, the adversary will place all the target event points along $\mathcal{P}_2$ to maximize the robot’s path. This part of the adversary’s construction of the polygon is shown in Fig. 9.

**Lemma 3.20.** Starting from the first target event point, the distance walked by the robot before it reaches $T$ is:

$$\leq 4.5d^2(3^{(d/2) - 1} - 1)(3^{(d/2) - 1} + 1) + \frac{4}{\delta^2}.$$

**Proof.** We refer to Fig. 12 for this proof. We assume that $q_1$ is the first blocked target event point. The subsequent blocked target event points are denoted as $q_2, q_3, \ldots, q_i$. The corresponding free target event points are $q'_1, q'_2, \ldots, q'_{i-1}$. The parts of the robot’s path which are inside the critical box are denoted by $\beta_2, \ldots, \beta_i$. The parts which are outside the critical box are denoted by $\gamma_1, \gamma_2, \ldots, \gamma_i$. The distance walked by the robot in going from $q_j$ to $q'_j$ is denoted by $\eta_j$, for $j = 1, \ldots, i$. Notice that, the adversary can place the first free target event point $q'_j$ at a distance which is 2 times the distance from $q_j$ to $T$ along the polygon boundary i.e., along $\mathcal{P}_1$. From Lemma 3.19, one part of $\mathcal{P}_1$ is inside the critical box (until the final source event point $p_i$) and the other part is $\sigma$. We represent the length of the first part as $\delta$. $\sigma$ consists of two parts, the first part is inside the critical box, i.e., $\sum_{j=1}^{i} \alpha_j$ and the second part is outside the critical box, i.e., $\sum_{j=1}^{i} \lambda_j$. So, the length of $\sigma$ (denoted by $|\sigma|$) is:
\[ |\sigma| \leq \sum_{j=1}^i \lambda_j + \sum_{j=1}^i \alpha_j = \left[ 2\alpha_1 + 2\alpha_{j-1} \sum_{k=0}^{1} 3^k + 2\alpha_{j-2} \sum_{k=0}^{2} 3^k + \cdots + 2\alpha_{i-1} \sum_{k=0}^{i-1} 3^k \right] + \sum_{j=1}^i \alpha_j \leq 2^{\frac{3i-1}{3-1}} - 1 \sum_{j=1}^i \alpha_j + \sum_{j=1}^i \alpha_j \leq d^2 3^{(d/2) - 1} \]

Notice that, \( \sum_{j=1}^i \alpha_j \leq \mathcal{O}d \) and there are \( i \leq \mathcal{O}/2 \leq d/2 \) pairs of blocked and free event point pairs. We represent the distance of \( T \) from \( q_1 \) as \( \beta_1 = |\sigma| + \delta \leq d^2 3^{(d/2) - 1} + \delta \). Arguing in a manner similar to Lemma 3.14, the point \( q'_1 \) can be placed at a distance of \( 2\beta_1 \). In other words, \( \gamma_1 \leq 2\beta_1 \).

It can be proved in a manner similar to Lemmas 3.15 and 3.16 that: \( \eta_j \leq 9(\beta_j + 3\beta_{j-1} + 3^2 \beta_{j-2} + \cdots + 3^{j-1} \beta_1) \). Also, \( \sum_{j=1}^i \eta_j \) is the total distance walked by the robot along the polygon boundary to reach \( T \) (from the first target event point).

\[
\sum_{j=1}^i \eta_j \leq 9 \times \frac{3^{j-1} - 1}{3-1} \sum_{j=1}^i \beta_j = 4.5(3^{(d/2) - 1} - 1) \left( \beta_1 + \sum_{j=2}^i \beta_j \right) \\
\leq 4.5(3^{(d/2) - 1} - 1) \left( d^2 3^{(d/2) - 1} + d^2 \right) \\
= 4.5d^2 \left( 3^{(d/2) - 1} - 1 \right) \left( 3^{(d/2) - 1} + 1 \right)
\]

The details of this calculation is similar to that of Lemma 3.17. We consume the term \( 9\delta \) in \( \sum_{j=2}^i \beta_j \leq d^2 \), since all these terms add up to \( \leq d^2 \). The total length of the free target segments is \( i \leq \mathcal{O}/2 \leq \frac{d}{2} \). So, the total distance walked by the robot from the first target event point until it reaches the target is: \( \leq 4.5d^2 \left( 3^{(d/2) - 1} - 1 \right) \left( 3^{(d/2) - 1} + 1 \right) + \frac{d}{2} \).

**Remark.** Similar to the remark after Lemma 3.13, it is easy to see that the robot’s path length can be increased only by a constant factor by adding a rake structure along the horizontal target line as well.

**Theorem 3.21.** The competitive ratio achieved by Algorithm Polygon_Search is: \( \leq 4.5d \left( 3^{(d/2) - 1} - 1 \right) \left( 3^{(d/2) - 1} + 2 \right) + \mathcal{O}(1) \).

**Proof.** The total distance walked by the robot is the sum of the quantities in the statements of Lemmas 3.17 and 3.20. The length of the shortest path is \( d \). Hence the competitive ratio of the strategy is the ratio of the distance walked by the robot and the length of the shortest path. From the remarks after Lemmas 3.13 and 3.20, a constant factor may be added to this. Hence we get the quantity in the statement of the theorem. \( \Box \)
Acknowledgements

The authors would like to thank three anonymous reviewers for many comments and suggestions that improved the presentation considerably. The first author’s research is partially supported by Western Australian Interactive Virtual Environments Centre (IVEC) and Australian Partnership in Advanced Computing (APAC).

References