Planning a Time-Minimal Motion Among Moving Obstacles

Kikuo Fujimura and Hanan Samet

Abstract. Motion planning for a point robot is studied in a time-varying environment. Each obstacle is a convex polygon that moves in a fixed direction at a constant speed. The point to be reached (referred to as the destination point) also moves along a known trajectory. The concept of "accessibility" from a point to a moving object is introduced, and is used to define a graph on a set of moving obstacles. If the point robot is able to move faster than any of the obstacles, then the graph exhibits an important property: a time-minimal motion is given as a sequence of edges in the graph. An algorithm is described for generating a time-minimal motion and its execution time is analyzed.

Key Words. Time-varying environments, Visibility graphs, Accessibility graphs, Motion planning.

1. Introduction. The motion-planning problem is one of finding a path that connects given start and destination points in an environment that contains a predefined set of obstacles so that the path is collision-free. The visibility graph [As], [Gh] has been an important combinatorial structure in planning paths among stationary polygonal obstacles [Lo], as well as in acquiring information about the environment while exploring it in two dimensions [Oo]. An important property of the visibility graph is that a shortest path is a finite sequence of edges of the graph. For surveys of research on motion planning, see [Wh] and [Ya].

In this paper we study the concept of accessibility, which is a generalization of the concept of visibility [Fu]. We make use of accessibility to represent moving objects for the purpose of planning the motion of a robot. The robot is assumed to move in a two-dimensional world in which polygonal obstacles, as well as the destination point, are in motion. The motions of the polygonal obstacles as well as the destination point are assumed to be known. The accessibility graph is a generalization of the visibility graph. Paths to the destination point are found as sequences of edges of the graph. In fact, when all the obstacles have zero velocity,
the accessibility graph becomes the visibility graph of these polygonal obstacles. Moreover, a time-minimal motion is represented as a sequence of edges of the accessibility graph.

Previous work dealing with moving obstacles includes the following: Reif and Sharir [Re] show that motion planning in a three-dimensional environment containing moving obstacles is PSPACE-hard given bounds on the robot's velocity, and NP-hard without such bounds. Canny and Reif [Ca] show that motion planning for a point in the plane with bounded velocity is NP-hard, even when the moving obstacles are convex polygons moving at constant linear velocity without rotation. These results indicate that motion planning with moving obstacles is computationally harder than motion planning with stationary obstacles. Nevertheless, there are some approaches to solving the problem. These approaches are successful in limited domains.

Kant and Zucker [Kan] decompose the problem of motion planning with moving obstacles into two parts. In the first part they plan a path among the stationary obstacles while ignoring the moving obstacles. In the second part a graph is used to define regions through which the robot may not pass when following the path computed in the first part. The positions of these regions influence the choice of the speed. Erdmann and Lozano-Perez [Er] make use of stacks (i.e., piles) of two-dimensional configuration spaces. These spaces are created each time some object changes its velocity. A path consists of a sequence of vertex-to-vertex transitions between two adjacent elements of the piles. None of these approaches deals with the case where the destination point can also be in motion.

In this paper we study the case in which the robot can move faster than any of the obstacles. This condition is the key that makes our problem tractable. Given a velocity (i.e., a speed and a direction in which to proceed), a robot can travel a certain distance before it meets with some obstacles (or it may never meet any obstacles). Let us call such a meeting point, if any, the collision point with respect to the velocity. The collision points around the start point, with respect to a certain speed, can be decomposed into subsets that we call collision fronts. They consist of a number of curves or line segments. Having computed collision fronts at the start point, the robot moves to an endpoint of one of the collision fronts, say $E$. At $E$ the robot is coincident with one of the vertices of the obstacles. Next we compute another collision front with $E$ as a start point. Now the robot moves from $E$ to one of the endpoints of the collision fronts that are generated about $E$. We repeat this process until we reach the destination point.

Dealing with moving obstacles and a moving destination point is of current interest in robotics. The ability to avoid moving obstacles leads to an increase in the mobility of a robot for navigation. It also results in higher productivity for factory manipulators. By allowing the destination point to move, we are able to consider a larger class of applications. For example, suppose that a robot arm must pick up an object that lies on a conveyor belt. As another example, consider an autonomous vehicle whose goal is to catch up with another vehicle that is in motion.
In a stationary environment a path is specified as a subset in the plane and the term "trajectory" is usually used to refer to velocity and acceleration information along a path. In a dynamic environment a collision-free path for a particular time period may not be collision-free for another time period. Therefore, a path must also be specified as a function of time. In this paper we use the term "motion" to mean a path with timing information, while the term "path" is used to specify a curve in space without timing information. Throughout our paper a goal point in motion is called a destination point, and the term point robot is used to refer to a point to be moved from the start point to the destination point. Note that we use the term "velocity" to include both the speed and direction of a motion, while we use the term "speed" to mean the magnitude of a motion.

The rest of this paper is organized as follows. In Section 2 we define the motions that obstacles are permitted to have, and introduce the concept of accessibility. Section 3 describes the procedure for finding a path and its time minimality. Section 4 contains an analysis of the execution time of the algorithm. Section 5 contains a few concluding remarks.

2. The Accessibility Graph. Throughout this paper O, G, and R are used to denote the start point, the destination point, and the point robot, respectively. First we define the motions of the obstacles, the point robot, and the destination point.

**Motion of an Obstacle.** An obstacle is a convex polygon that moves in a fixed direction at a constant speed. We call such a straight motion a movement. To simplify the explanation we treat the line segments (edges) that constitute polygons as the basic units of our discussion. A movement is defined by a tuple \((L, d_L, v_L)\) that represents the motion of line segment \(L\) in direction \(d_L\) at speed \(v_L\). We consider an environment that contains a finite set of movements,

\[ M = \{M_1, M_2, \ldots, M_n\}, \]

where each \(M_i\) represents a movement as defined above. Note that \(M\) corresponds to the motions of all the edges in the environment, not just those in one polygon. Hence, if a polygon \(P_i\) consists of \(l_i\) edges (thus vertices), we have \(n = l_1 + l_2 + \cdots + l_k\), where \(k\) is the number of polygons in the environment.

**Motion of a Point Robot.** After leaving the start point at the start time, \(R\) can have any motion as long as its speed does not exceed a given maximum speed and does not pass through the interiors of obstacles. We assume that the maximum speed of \(R\) is greater than that of any of the obstacles, and is also greater than that of the destination point.
MOTION OF A DESTINATION POINT. The motion of the destination point consists of a finite series of steps. Within a step, the destination point moves in a constant direction at a constant speed. We assume that the number of steps is bounded.4

Now let us define the concepts of accessibility, collision front, accessible vertex set, and accessibility graph.

ACCESSIBILITY. Consider a set of movements \( M = \{ M_1, M_2, \ldots, M_n \} \) and \( G \), the destination point. Let \( R \) be a point robot located initially at \( O \) at time \( t_0 \). Suppose that \( R \) starts moving at time \( t_0 \) at a speed \( v \). After \( R \) starts moving, it moves in a fixed direction. A point \( V \) (is either the destination point or a point on a polygonal obstacle) is said to be accessible from \( O \) if there exists a direction of the motion of \( R \) such that \( R \) meets \( V \) without prior interception by any other movement. We say that \( V \) and \( R \) meet if there exists a location \( X \) through which both \( V \) and \( R \) pass at the same time \( t \), where \( t_0 < t \). The location \( X \) is called an accessible point of \( V \). The time \( t \) is called the accessible time of \( X \) with respect to \( V \) and is denoted by \( t(X) \). Observe that the accessible point of a particular point \( V \) varies for different values of the speed and the initial location of \( R \). Also note that if a stationary point \( V \) (\( V \) is either a point on a stationary obstacle or a stationary destination point) is accessible, then its accessible point is \( V \) itself.

COLLISION FRONT. Consider an environment that contains one movement, say \((L, d_L, v_L)\). Let \( V_a \) and \( V_b \) be the two endpoints of \( L \), and let \( P_a \) and \( P_b \) be accessible points corresponding to \( V_a \) and \( V_b \), with respect to \( R \)'s initial location \( O \), start time \( t_0 \), and speed \( v \). The set of accessible points corresponding to all points in \( L \) forms a segment. As is shown later in Section 4, this segment takes the form of either a straight line or a quadratic curve. We call this segment a collision front of \( L \) (with respect to \( O, t_0 \), and \( v \)). Figure 1 contains an example of the collision front. It can be shown that \( P_a \) and \( P_b \) are the two endpoints of the collision front. For an environment that contains more than one movement, there can be more than one collision front, each of which corresponds to some movement. In this case,

---

4 This condition can be relaxed. See Section 5 for more details.
however, it is possible that only a part of an obstacle is accessible. Notice that if two moving obstacles (e.g., edges) do not collide, then the corresponding collision fronts do not intersect. The characterization of the collision front is as follows. Suppose that \( R \) departs \( O \) at time \( t_0 \) and keeps moving at a constant speed \( v \) along a ray, say \( l \), emanating from \( O \). The point robot \( R \) does not meet any of the obstacles in motion if and only if \( l \) does not intersect any of the collision fronts. The collision front of a stationary edge \( L \) is \( L \) itself or consists of subsets of \( L \).

**Accessible Vertex Set.** Let \( VS \) be the set consisting of the destination point and the vertices of the polygonal obstacles in motion \( M \), and let \( O, t_0, v \) be as in the definition of accessibility. The set of accessible points corresponding to elements of \( VS \) with respect to \( O, t_0, v \) is called the accessible vertex set and denoted as \( AVS(M, O, t_0, v) \). Since some vertices in \( VS \) may not be accessible from \( O \), the size of \( AVS \) is at most \( |VS| \).

**Accessibility Graph.** Let \( O \) be the start point (i.e., the point at which point robot \( R \) is initially found), let \( t_0 \) be the start time, let \( v \) be the start speed, and let \( G \) be the destination point. With each accessible point \( X \) corresponding to a point \( V \), we also associate a time value (i.e., timestamp), say \( t(X) \), to denote \( X \)'s accessible time with respect to \( V \). We define a directed graph called the accessibility graph, or \( AG(M, O, G, t_0, v) \), by the following construction:

1. Insert \( O \) in the vertex set of \( AG \), and set its default accessible time to \( t_0 \).
2. For every newly added vertex \( U \) in the set of vertices in \( AG \), consider the accessible vertex set with \( U \) as the initial point, i.e., \( AVS(M, U, t(U), v) \). Insert the elements of this \( AVS \) in the vertex set of \( AG \), and the edges from \( U \) to these points in the edge set of \( AG \).

The graph is a (directed) tree\(^5\) rooted at \( O \). Note that for a given set of \( M, O \), and \( G, AG \) varies for different values of \( t_0 \) and \( v \). This graph is infinite according to our definition. However, we can make it finite by adopting the rule that when a vertex of an input polygon is used more than once to define vertices of \( AG \), then the instance with the youngest (i.e., earliest) accessible time is the one that is retained. Also note that the vertices of \( AG \) are the accessible points and not the original points on the obstacles.

Figures 2(a) and (b) demonstrate how to construct an accessibility graph. Points \( S \) and \( G \) are the starting and destination points, respectively. Triangle \( ABC \) and rectangle \( DEFH \) are obstacles moving in the indicated directions. First, \( S \) is inserted in the vertex set of the accessibility graph (step 1). Vertices \( B, C, \) and \( F \) are accessible from \( S \) at \( P, Q, \) and \( U \), respectively, but \( E \) is not accessible because obstacle \( ABC \) is in the way. Therefore, \( AVS(M, S, t(S), v) = \{ P, Q, U \} \). Points \( P, Q, \) and \( U \) are inserted in the vertex set of the accessibility graph. Accordingly, \( SP, SQ, \) and \( SU \) are inserted in the edge set of the accessibility graph (step 2).

\(^5\) In degenerate cases the graph might not be a tree but only a DAG (e.g., when there are two shortest paths to a vertex.)
Fig. 2. Construction of an accessibility graph. (a) Insert the start point in the vertex set of $AG$. (b) The AVS with $Q$ as the start point. Points $Y$ and $W$ will not be included in the accessibility graph when the finite version of the definition of the accessibility graph is used. (c) The result of applying procedure PlanMotion to (a).

let us take $Q$ as a newly added vertex in step 2 of the construction (Figure 2(b)). Vertices $A$, $B$, $E$, and $F$ of the obstacles are accessible at locations $X$, $Y$, $V$, and $W$, respectively, i.e., $AVS(M, Q, t(Q), v) = \{X, Y, V, W\}$. Thus, $X$, $Y$, $V$, and $W$ are inserted in the vertex set of the graph, and $QX$, $QY$, $QV$, and $QW$ are inserted in the edge set of the graph. This process is applied to other vertices such as $P$, $U$, etc.

Note that according to the finite definition of the accessibility graph, points $Y$ and $W$ (Figure 2(b)) will not appear in the graph since $P$ and $U$ (the accessible
points of $B$ and $F$ from $S$, respectively, as can be seen in Figure 2(a) have younger accessible times when they are reached from $S$.

The accessibility graph is related to the visibility graph in the following manner. Consider a stationary set of movements $M$ whose elements do not overlap. Construct the (infinite) accessibility graph for $M$, and make all directed edges in $AG$ undirected and remove the timestamps at all the vertices. The resulting graph is the visibility graph for $M$. Since we will need only the finite part of the accessibility graph in the subsequent sections, we retain its finite formulation.

3. Time-Minimal Motion. In this section we first describe a procedure that finds a motion to the destination point using $AG$ as defined in the previous section. Next we prove that the motion found by the procedure is time-minimal. In this process we use a priority queue of vertices where the accessible time of each vertex serves as the vertex's priority.

Procedure PlanMotion:

1. Insert the starting vertex $O$ into the queue.
2. Remove a vertex, say $V$, from the queue whose associated accessible time is the youngest.
3. If vertex $V$ in step 2 is the destination point $G$, then report the motion and exit; otherwise, insert all the vertices that are adjacent to $V$ in $AG$ into the queue, and repeat steps 2 and 3.

Step 3 needs some care. An element of $AGS$ is inserted into the queue when the corresponding vertex has not been seen before or it has a younger accessible time than the one (for the same corresponding vertex) that is already in the queue in which case the old element is deleted. When a new element has an older accessible time than the one for the same vertex that is already in the queue, the new element is not added. As a result, the size of the queue never exceeds the sum of the total number of vertices of the input polygons plus the destination point. By constructing the graph in this way, we obtain the finite version of the accessibility graph defined in Section 2.

Figure 2(c) is an example of planning a motion using procedure PlanMotion. Points $S$ and $G$ are the start and destination points, respectively. Triangle $ABC$ is an obstacle moving toward the right, and rectangle $DEFH$ is an obstacle moving toward the left. Points $Q$, $V$, and $T$ are the accessible points of $C$ from $S$, of $E$ from $Q$, and of $D$ from $V$, respectively. The dashed objects show the locations of the corresponding obstacles at the indicated times. Note that while the robot is moving between $V$ and $T$, it is coincident with some point on edge $DE$ of obstacle $DEFH$. The sequence of line segments $(SQ, QV, VT, TG)$ constitutes the final motion.

In the rest of this section, let $v_{max}$ denote $R$'s maximum speed, and let $O$ and $G$ be the start and destination points, respectively. We subdivide the motions from $O$ to $G$ into two groups, i.e.:
Group 1. Motions whose initial segments (i.e., start at \(O\)) are traversed at speed \(v_{\text{max}}\) in a constant direction through a point in \(AVS(M, O, t_0, v_{\text{max}})\), say \(P\). The motion from \(P\) is arbitrary but must eventually reach \(G\).

Group 2. The other motions.

It is assumed that speeds along motions in both groups do not exceed \(v_{\text{max}}\). In the rest of this paper we assume the following two conditions:

1. The maximum speed of the point robot \(v_{\text{max}}\) is greater than the speed of any of the obstacles.
2. The obstacles never collide with each other nor do they collide with the destination point.

**Proposition 1.** Given a set of movements \(M\) and a destination point \(G\), suppose that point \(R\) and vertex \(V\), the common endpoint of edges \(UV\) and \(VW\) of the same polygonal obstacle, are at location \(X\) at time \(t_1\). Let \(M_i\) and \(M_j\) be the movements of edges \(UV\) and \(VW\), respectively, and let \(M_k\) be the movement corresponding to the motion of the new edge \(UW\). If there exists a motion starting from \(X\) at time \(t_1\) and terminating at \(G\) at time \(t_2\) (> \(t_1\)) in \(M - \{M_i, M_j\} + \{M_k\}\) (i.e., an environment formed by removing the triangle \(UVW\) from the polygonal obstacle, but leaving edge \(UW\)), then there exists a motion in \(M\), starting from \(X\) at time \(t_1\) and terminating at \(G\) at time \(t_2\).

**Proof.** Let \(\alpha\) be a motion from \(X\) to \(G\) in \(M - \{M_i, M_j\} + \{M_k\}\). There are two cases depending on whether or not \(\alpha\) collides with the interior of the moving triangle \(UVW\).

**Case 1: Motion \(\alpha\) Does Not Collide with the Triangle Interior.** By construction, \(\alpha\) is collision-free in \(M\).

**Case 2: Motion \(\alpha\) Collides with the Triangle Interior.** We construct an alternative motion \(\beta\) such that \(\beta\) is collision-free in \(M\) in the following manner. Let \(Y\) be the location of the occurrence of the last collision of \(\alpha\) with the moving triangle \(UVW\) (see Figure 3). Without loss of generality, let \(L\) denote the movement (i.e., \(M_i\) or \(M_j\)) with which \(\alpha\) collides last. (Point \(Y\) can coincide with an endpoint of \(L\).) Starting at time \(t_1\), define \(\beta\) as the straight-line motion followed by \(R\) from \(X\) to \(Y\) with a constant speed such that \(\beta\) reaches \(Y\) exactly at the same time as \(\alpha\) reaches \(Y\) for the last time. This means that until reaching \(Y\), \(\beta\) is always coincident with some point on \(L\). The length of this motion is shorter than the corresponding segment of \(\alpha\). To see this, note that the shortest distance between two points is a straight line. After \(Y, R\) follows the remaining part of motion \(\alpha\). The point \(R\) reaches \(G\) at the same time regardless of whether motion \(\alpha\) or \(\beta\) is used. Along \(\beta\), \(R\) moves at a speed slower than \(v_{\text{max}}\) from \(X\) to \(Y\). To see this, consider two motions from \(X\) to \(Y\), namely, \(\alpha\) and \(\beta\), both starting and arriving at the same time. Since \(\beta\) is shorter than \(\alpha\), the speed of \(R\) between \(X\) and \(Y\) must be less than \(v_{\text{max}}\).
As a special case of this proposition, we consider a case in which an obstacle is just a single edge. We have the following proposition; its proof is similar to the proof of Proposition 1.

**Proposition 1'**. Assume the same environment as in Proposition 1 except that UV, with movement M, is a single edge obstacle. Suppose that at time $t_1$, point R and vertex V are at location X. If there exists a motion starting from X at time $t_1$ and terminating at G at time $t_2$ in $M - \{M\}$, then there exists a motion in M starting from X at time $t_1$ and terminating at G at time $t_2$.

**Proposition 2**. Let $\sigma_1$ be a motion of a point robot, say $R_1$, consisting of two linear movements. Let the first movement be from O to A at a constant speed $v_1$ along a line segment OA and let the second movement be from A to infinity at a constant speed $v_2$. Let $v_1 < v_{\text{max}}$ and $v_2 < v_{\text{max}}$. For any point P on $\sigma_1$, there exists a motion of a point robot, say $R_2$, from O to P such that $R_2$ leaves O at the same time as $R_1$, $R_2$ moves along a straight line OP at a speed $v'$ ($v' < v_{\text{max}}$), and $R_2$ arrives at P either before $R_1$ or at the same time.

**Proof**. Choose $v' = \max\{v_1, v_2\}$. This proposition holds by applying the triangle inequality to OAP (Figure 4). \hfill \Box

**Proposition 3**. Suppose that G is not accessible from O. Let $\pi_2$ be a collision-free motion from O to G. Let $\pi_2$ be in Group 2. There exists a collision-free motion from O to G in Group 1, say $\pi_1$, that is identical to $\pi_2$ starting at some point J. The speed along some portion of $\pi_1$ before J is less than $v_{\text{max}}$. 
Proof. In the first part of the proof let us assume that there is only one collision front about \( O \) for \( v = v_{\text{max}} \). Let \( V_1 \) and \( V_2 \) denote the two endpoints of the collision front. Let \( v_{\text{a}} \) denote the maximum speed of the obstacles in the scene.

The outline of the proof is as follows: as candidates for \( \pi_1 \) in Proposition 3, we construct two motions, \( \tau_1 \) and \( \tau_2 \), that start with segments \( OV_1 \) and \( OV_2 \), respectively, each moving at speed \( v_{\text{max}} \). Next, each of them moves at a speed infinitesimally less than \( v_{\text{max}} \) (i.e., \( v_{\text{max}} - \varepsilon \) for a small \( \varepsilon \); the value of \( \varepsilon \) remains to be determined) and heads for the earliest point \( J \) on \( \pi_2 \) that they can reach.

In the following we show that it is possible to choose one of \( \tau_1 \) or \( \tau_2 \) as \( \pi_1 \). To carry out the proof, we classify the collision front into three types: receding obstacle, proceeding obstacle, and degenerate obstacle, as defined below. To see this classification, consider a point robot that moves straight from \( O \) to a point in the collision front \( V_1 V_2 \) at speed \( v_{\text{max}} \), and let the point robot stand still at that spot for a brief moment, while the obstacle continues moving.

Type 1 (receding obstacle): The point robot will not collide with the obstacle.
Type 2 (proceeding obstacle): The point robot will collide with the obstacle.
Type 3 (degenerate obstacle): Until the point robot stops moving, it is always collinear with the obstacle.

Figures 5(a)–(c) contain examples of a receding obstacle, a proceeding obstacle, and a degenerate obstacle, respectively. The type can be determined by the relation between the initial location of the obstacle (line segment) and \( O \). Let \( l \) be the line in motion that is the extension of the obstacle. When \( l \) intersects \( O \) after it starts moving, \( l \) is a proceeding obstacle (type 2). When it does not, it is a receding obstacle (type 1). When \( O \) is on \( l \), it is of type 3.
Fig. 5. Three types of obstacles: (a) receding, (b) proceeding, and (c) degenerate.

Case 1: Receding Obstacle.

Proof. We consider two auxiliary motions, σ₁ and σ₂, starting from V₁ at time t(V₁) and from V₂ at time t(V₂), respectively (see Figure 6(a)). Motions σ₁ and σ₂ move at constant speeds v₁ and v₂, respectively, where vᵢ < v_{max} (i = 1, 2), and they are always coincident with some point in L. Let W be the point at which σ₁ and σ₂ meet (Figure 6(a)). Choose vᵢ so that the ray OW crosses the collision front. Let l₁ and l₂ be the two rays emanating from V₁ and V₂ that are extensions of rays OW₁ and OW₂, respectively. An open-ended serial chain of line segments l₁, V₁W, W₂, and l₂ partitions the plane into two regions. Let A be the region containing O and let B be its complement.
Fig. 6. A receding obstacle for the proof of Proposition 3.

First we show that if there are no moving obstacles besides \( L \), then one of \( \tau_1 \) or \( \tau_2 \) will reach \( \pi_2 \) in region \( A \). There are two cases depending on whether or not \( \pi_2 \) crosses the border of the two regions. In the first case we assume that \( \pi_2 \) crosses the border of the two regions and let \( Z \) be the first point at which such a crossing occurs. We show that either \( \tau_1 \) or \( \tau_2 \) can reach \( \pi_2 \) before \( \pi_2 \) enters region \( B \) (i.e., \( J \) occurs at or before \( Z \) along \( \pi_2 \)).

Suppose that \( Z \) lies on \( WW'_1 \) or \( WW'_2 \) (see Figure 6(a)). Let \( Y \) be the point at which \( \pi_2 \) crosses the collision front \( V_1V_2 \). Motion \( \pi_2 \) goes through \( Y \) at the same time
as or after $L$ has gone through $Y$, because $Y$ is an accessible point and $\pi_2$ is a Group 2 motion. Since it is impossible for $\pi_2$ to penetrate $L$ between $Y$ and $Z$, $\pi_2$ also goes through $Z$ at the same time as or after $L$ has gone through $Z$. Now, recall that segment $V_1W$ (or $V_2W$) is a trajectory for a point robot whose motion is always coincident with some point in $L$. Therefore, the point robot goes through $Z$ at the same time as or earlier than $\pi_2$ goes through $Z$. This implies that $\varepsilon$ can be chosen so that one of $\tau_1$ or $\tau_2$ will reach $\pi_2$ in region $A$. For the case that $Z$ lies on $l_1$ (or $l_2$ as in Figure 6(b)), it should be clear that, by choosing $\varepsilon$ to be sufficiently small, $\tau_1$ (or $\tau_2$) with a speed $v_{\max} - \varepsilon$ will reach $\pi_2$ in region $A$.

In the second case we assume that $\pi_2$ does not cross the border of the two regions. If $\pi_2$ reaches the destination point without crossing the border, then either $\pi_1$ (one of $\tau_1$ or $\tau_2$ depending on whether we are departing from $V_1$ or $V_2$) can reach the destination point before $\pi_2$ does, or $\pi_1$ can reach $\pi_2$ before $\pi_2$ reaches the destination point. This can be shown as follows.\(^6\) Let $K$ be the point at which $G$ crosses the border (Figure 6(c)). Let $K$ lie on $l_2$ as Figure 6(c) (proofs for the other cases are analogous). Let $X$ be the point at which $\pi_2$ meets $G$. Consider a point robot that departs $V_2$ at time $t(V_2)$ and moves straight to $X$ at speed $v_{\max} - \varepsilon$. If the point robot arrives at $X$ before $\pi_2$, then we are done since it implies that there is a motion from $V_2$ at $v_{\max} - \varepsilon$ that reaches $\pi_2$ before $\pi_2$ reaches $X$. Otherwise, we can show that there exists a location (say $N$) between $K$ and $X$ at which $\pi_1$ can meet $G$ as follows. Consider a motion that departs $V_2$ at time $t(V_2)$ with a speed $v_{\max} - \varepsilon$. This motion is directed toward $K$ or $X$. When this motion is aimed toward $K$, it can reach $K$ before $G$ reaches $K$. When this motion is directed toward $X$, it reaches $X$ only after $G$ has reached $X$. This means that there exists a point, say $N$, between $K$ and $X$ having the property that if the motion is directed toward $N$, then it will reach $N$ just as $G$ passes $N$. We choose this motion as $\tau_2$ (a candidate for $\pi_1$).

Now we must show that one of $\tau_1$ or $\tau_2$ is indeed collision-free, even when there are other obstacles present besides $L$.

At this point let us describe how we choose a speed along $\tau_1$ or $\tau_2$ (i.e., the choice of $v_{\max} - \varepsilon$). Consider all obstacles that exit from region $B$ into region $A$ and that possibly intercept $\tau_1$ or $\tau_2$. Since $V_1V_2$ is the only collision front in the scene, the motions of the obstacles must cross $l_1$, $l_2$, $V_1W$, or $V_2W$ at some point. Moreover, the crossing times have the following characteristic. For any obstacle that possibly intercepts $\tau_1$ or $\tau_2$, let $Q$ be the last crossing point between regions $A$ and $B$, before it intercepts $\tau_1$ or $\tau_2$. When $Q$ lies on $l_1$ (or $l_2$), the crossing time must be strictly later than the time at which a point that moves along $l_1$ (or $l_2$) at $v_{\max}$ departing $O$ at time $t_0$ would reach $Q$. Otherwise, it contradicts our assumption that there is only one collision front. When $Q$ lies on $V_1W$ (or $V_2W$), the crossing time must be strictly later than the time at which motion $\sigma_1$ (or $\sigma_2$) passes $Q$, since the obstacles cannot collide (in this case this would have required the obstacle to collide with $L$).

This implies that we can always construct a motion\(^7\) from $V_1$ (or $V_2$) at a speed

\(^6\)The following proof does not require that the robot move faster than the destination point.

\(^7\)This motion itself may or may not be collision-free. However, it does not affect our proof.
less than $v_{\text{max}}$ that reaches any of those obstacles when it crosses $l_1$ or $V_1 W (l_2$ or $V_2 W)$. Let $v_\rho$ be the speed of the fastest of all such motions. Let $v_{\text{max}} - \varepsilon = \max\{v_\rho, v_{c_1}, v_{c_2}, v_\delta\}$. Recall that $v_\rho$ is the maximum speed of the obstacles in the scene. Then by applying Proposition 2 (letting the appropriate one of $V_1$ and $V_2$ correspond to $O$, and letting $Q$ correspond to $A$), $\tau_1$ will not collide with any obstacle crossing $l_1$ or $V_1 W$, and $\tau_2$ will not collide with any obstacle crossing $l_2$ or $V_2 W$. For $\pi_1$ of Proposition 3, choose the one of $\tau_1$ or $\tau_2$ that reaches $\pi_2$ earlier.

We prove that motion $\pi_1$, chosen as such, is collision-free. Without loss of generality, let $\tau_1$ be chosen as $\pi_1$. By construction, $\tau_1$ does not collide with any obstacle that crosses $l_1$ or $V_1 W$. Now, suppose that $\tau_1$ collides with an obstacle crossing $l_2$ or $V_2 W$. Figure 6(d) shows such a scene, where $V_1$ and $V_2$ are the accessible points of $P_1$ and $P_2$, respectively. Let $C$ be an obstacle that intercepts motion $\tau_1$ at $X$. Let $Q$ be a point at which $C$ crosses $V_2 W$. Note that the time at which $C$ passes $Q$ must be strictly later than the time at which motion $\sigma_2$ passes $Q$. Applying Proposition 2 to $V_2 Q X$ (letting $V_2$ and $Q$ correspond to $O$ and $A$, respectively) as well as to $V_2 X J$ (letting $V_2$ and $X$ correspond to $O$ and $A$, respectively), it can be shown that there is a motion that will reach $J$ before $\tau_1$. This implies that $\tau_2$ can reach $\pi_2$ earlier than $\tau_1$. This contradicts the choice of $\pi_1$. (The same argument works if $Q$ is on $l_2$).

**Case 2: Proceeding Obstacle.**

**Proof.** The proof for this case is analogous to the proof for receding obstacles.

This time the plane is divided into two regions by $l_1$, $l_2$, and the collision front. Let $A$ be the region containing $O$ and let $B$ be its complement. Let $Y$ be the first point at which $\pi_2$ crosses the border. We consider only the case in which $Y$ lies on the collision front. The other cases (i.e., the case in which $Y$ lies on either $l_1$ or $l_2$ and the case in which $\pi_2$ reaches the destination point without crossing the border) are the same as the corresponding parts of the proof for receding obstacles.

We consider motions $\sigma_1$ and $\sigma_2$ at a speed $v_\rho$ where $v_\rho < v_c < v_{\text{max}}$. Let $W$ be the point at which $\sigma_1$ and $\sigma_2$ meet. It should be clear that $\pi_2$ must cross either $V_1 W$ or $V_2 W$ (as in Figure 7(a)) before $\pi_2$ reaches $Y$. Note that when the trajectory of $L$ contains $O$ (Figure 7(b)), it is impossible for $\pi_2$ to reach the destination point without placing itself outside $L$’s trajectory (even just momentarily), since $\pi_2$ cannot penetrate $L$.

Let $Z$ be the point at which $\pi_2$ crosses $V_1 W$ or $V_2 W$ for the last time before $\pi_2$ passes $Y$. Note that portion $ZY$ of $\pi_2$ is entirely inside the trajectory swept by $L$, and that $\pi_2$ passes $Y$ after $L$ has passed it. Motion $\pi_2$ must pass $Z$ only after $L$ has passed $Z$, since $\pi_2$ cannot penetrate $L$. This means that $\pi_2$ passes $Z$ later than the time at which $\sigma_1$ or $\sigma_2$ passes $Z$. For the same reason as in the case of receding obstacles, $\tau_1$ departing from $V_1$ or $\tau_2$ departing from $V_2$ can reach $\pi_2$ before $\pi_2$ reaches $Z$. As $\pi_1$, pick one of $\tau_1$ or $\tau_2$ that reaches $\pi_2$ earlier. Motion $\pi_1$ is shown to be collision-free as in the case of a receding obstacle.

**Case 3: Degenerate Obstacle.** In this case the collision front is collinear with $O$. The destination point $G$ must be on the extension of the collision front (otherwise,
$G$ is accessible from $O$. For a sufficiently small $\alpha$, it is clear that $\tau_1$ or $\tau_2$ will reach $\pi_2$ before $\pi_2$ reaches $G$.

Next we consider the case in which there is more than one collision front generated at $O$. If there are $n$ collision fronts, then there are at most $n$ regions designated as $B$. Let us use $B_i$ to denote such a region behind collision front $CF_i$ (Figure 8). There are at most $2n$ choices for motion $\pi_1$—two for each collision front. For each of the collision fronts, $CF_i$, there is a range of speed values $[v_{\text{max}} - \varepsilon_i, v_{\text{max}}]$ such that the corresponding motion $\pi_{1i}$ joins motion $\pi_2$ at point $J_i$. Pick a value of the speed $v_x$ such that $v_x$ is in each of $[v_{\text{max}} - \varepsilon_i, v_{\text{max}}]$. Choose a motion $\pi_{1i}$ such that $J_i$ is reached at the earliest time of all $J_i$'s. Now
we must show that this motion is collision-free. Suppose that it is not. In this case
the obstacle intercepting $\pi_{ij}$ must come from some $B_j$ region ($j \neq i$). Then it can
be shown by applying Proposition 2 twice that a motion $\pi_{ij}$ (a candidate for $\pi_i$
departing from an endpoint of $CF_j$) will join motion $\pi_2$ at an earlier time. In
Figure 8 let $Z$ be a point at which $C$ crosses the boundary between $A_i$ and $B_j$, let
$V_j$ be an endpoint of collision front $CF_j$. Applying Proposition 2 on $V_jZX$ and
$V_jXJ_i$, we can construct a motion that reaches $J_i$ before $\pi_{1i}$. This means that $\pi_{1j}$
can reach $\pi_2$ earlier than $\pi_{1i}$ (i.e., $J_i$ occurs before $J_i$ along $\pi_2$), which contradicts
the choice of $\pi_i$.

**Theorem.** A time-minimal motion is given as a sequence of edges of

$$AG(M, O, G, t_0, v_{\max}).$$

As a result of this theorem, point robot $R$ moves at the maximum speed along
a time-minimal motion. In other words, if a motion contains some portion during
which $R$ moves at a slower speed, then that motion is not time-minimal. We prove
the theorem by induction on $n$, the total number of movements in the given
environment.

**Proof (base step).** In the base case (i.e., $n = 0$) $G$ is accessible from $O$, the start
point, since there are no obstacles (stationary or moving) in the scene. Recall that
we treat stationary obstacles as obstacles with zero velocity. Thus, $G$'s accessible
point, $X$, is in $AVS(M, O, t_0, v_{\max})$, and the procedure terminates with motion $OX$
along which $R$ moves at $v_{\max}$. Now we show that this motion $OX$ is time-minimal.
Suppose that it is not. Let $\gamma$ be a time-minimal motion and let $Y$ be the point at
which $R$ reaches $G$. Motion $\gamma$ can be extended to point $X$ such that $R$ is always
coincident with $G$ from $Y$ to $X$. Now there are two motions from $O$ to $X$, i.e., $OX$
and $\gamma$ plus its extension, both of which reach $X$ at the same time. This is a
contradiction since $R$ moves at $v_{\max}$ along straight line $OX$.

(Inductive step) Next we consider an environment that contains $n$ movements.
We assume that the theorem holds for an environment with $n - 1$ movements. If
$G$ is accessible from $O$, then the theorem obviously holds as in the case of $n = 0$.
Therefore, assume that $G$ is not accessible from $O$. The theorem follows from the
following two properties proved below. Let $V$ be a point in $AVS(M, O, t_0, v_{\max})$.
First, we prove that a time-minimal motion from $V$ to $G$ in $M$ consists of a
sequence of edges in $AG(M, V, G, t(V), v_{\max})$. Second, we prove that for any motion
in Group 2, a better motion in Group 1 exists. Thus a motion in Group 2 is never
time-minimal.

**Proof of the First Property.** Let $V$ be an accessible point (from $O$) of a vertex in
movement $M_i$. Note that at time $t(V)$, $R$ is on a vertex of some polygon. Suppose
that $L_i$ and $L_j$ are the two edges that are incident on the vertex. We remove $L_i$ and
$L_j$ from the polygon and introduce a new edge $L_k$ that connects the two endpoints
of $L_i$ and $L_j$ that are not incident at the vertex. This results in a polygon that
has one less edge than before. By the induction hypothesis, a time-minimal motion $\gamma$
exists from $V$ to $G$ in $M - \{M_i, M_j\} + \{M_k\}$, consisting of the edges of
$AG(M - \{M_r, M_j\} + \{M_k\}, V, G, t(V), \nu_{max})$, where $M_r$, $M_j$, and $M_k$ are the movements of $L_r, L_j$, and $L_k$, respectively. We show that this motion $\gamma$ is also collision-free in $M$. In other words, $\gamma$ does not collide with $M_r$ or $M_j$.

Assume that $\gamma$ collides with $M_r$. Since $R$ is located at $V$, we can use Proposition 1 to construct a motion $\gamma'$ in $M$ that terminates at $G$ at the same time as $\gamma$ in $M - \{M_r, M_j\} + \{M_k\}$. In case $M_j$ is a single edge, then Proposition 1’ is applied and $M - \{M_k\}$ is used in place of $M - \{M_r, M_j\} + \{M_k\}$ in the following argument. Motion $\gamma'$ must be one of the time-minimal motions from $V$ to $G$ in $M - \{M_r, M_j\} + \{M_k\}$, as it terminates at $G$ at the same time as $\gamma$. However, $\gamma'$ contains a path segment along which $R$ moves at a slower speed than $\nu_{max}$ (as noted in the proof of Proposition 1). Therefore, by the induction hypothesis, $\gamma'$ cannot be a time-minimal motion from $V$ to $G$ in $M - \{M_r, M_j\} + \{M_k\}$. Since $\gamma$ and $\gamma'$ terminate at $G$ at the same time, $\gamma$ is also not a time-minimal motion, which is a contradiction (recall our initial assumption was that $\gamma$ is time-minimal). Thus, motion $\gamma$ does not collide with $M_r$ and is a collision-free time-minimal motion from $V$ to $G$ in $M$ as well. This can be seen by noting that if a collision-free motion is time-minimal in $M - \{M_r, M_j\} + \{M_k\}$, then it is also time-minimal in $M$.

Let $\pi$ be a motion from $V$ to $G$ in $M$ that contains a segment during which $R$ moves at a speed less than $\nu_{max}$. We now show that $\pi$ will terminate at $G$ at a time later than $\gamma$. By the induction hypothesis, in $M - \{M_r, M_j\} + \{M_k\}$, $\pi$ terminates at $G$ later than $\gamma$. Since $\gamma$ is also a motion in $M$, as we showed above, $\pi$ terminates at $G$ later than $\gamma$ in $M$, and thus $\pi$ is not time-minimal in $M$. Thus our first property has been proved.

**Proof of the Second Property.** Assume that there exists a motion $\alpha$ in Group 2. We show that there exists a motion in Group 1 that reaches $G$ earlier than $\alpha$. Let $\beta$ be a motion from $O$ to $G$ constructed in the following manner. After leaving $O$ at time $t_0$, $\beta$ moves to one of the vertices in $AVS(M, O, t_0, \nu_{max})$ (let us call this vertex $V$) at speed $\nu_{max}$, and then turns to rejoin motion $\alpha$ at a speed $\nu$ less than $\nu_{max}$. Proposition 3 assures that this rendezvous is possible. After rejoining $\alpha$, the remaining parts of motion $\beta$ are identical to motion $\alpha$, and $\beta$ terminates at $G$ at the same time as $\alpha$. On the other hand, in every time-minimal motion $\gamma$ from $V$ to $G$ in $M$, every motion segment has speed $\nu_{max}$, as described in the first property. Since $\beta$ has a motion segment along which $R$ moves at a slower speed than $\nu_{max}$, $\beta$ is not time-minimal from $V$ to $G$ and $\beta$ terminates at $G$ later than $\gamma$. This implies that we can construct a motion in $M$ that terminates at $G$ earlier than $\alpha$ by concatenating motions $\beta$ and $\gamma$ at $V$. Thus our second property has been proved.

From the first property, the time-minimal motion from $V$ to $G$ consists of edges in $AG(M, V, G, t(V), \nu_{max})$, which is a subgraph of $AG(M, O, G, t_0, \nu_{max})$. From the second property, a time-minimal motion from $O$ to $G$ must contain edge $OV$ (for some $V$); this is an edge of $AG(M, O, G, t_0, \nu_{max})$. Thus the time-minimal motion is a sequence of edges of $AG(M, O, G, t_0, \nu_{max})$.

### 4. Execution Time.

We now analyze the time required to generate a collision-free motion using the algorithm described in Section 3. First we show that the shape of the collision front for a single moving segment is either a parabola, hyperbola,
or ellipse. Without loss of generality, we consider a line segment $L$ lying parallel to the $y$-axis. Let $(x_0, y_0)$ be the location of the lower one of $L$’s endpoints at the start time, let $l$ be the length of $L$, let $v_L$ be the speed of $L$, let $v_{\text{max}}$ be the speed of the robot, and let $\theta$ be the angle formed by the direction of the movement and the $x$-axis (Figure 9). Suppose that a point $P$ at $(x_0, Y)$ in $L$ is accessible from the origin $(0, 0)$ at point $A$ at $(x, y)$. Coordinates $x$ and $y$ must satisfy

1. \[ t(A) = \frac{\sqrt{x^2 + y^2}}{v_{\text{max}}} = \frac{\sqrt{(x - x_0)^2 + (y - Y)^2}}{v_L}, \]

2. \[ y = (x - x_0) \tan \theta + Y, \]

3. \[ y_0 \leq Y \leq y_0 + l. \]

The points $(x, y)$ that satisfy (1) and (2) form the collision front. These equations define a quadratic relationship, i.e., the collision front lies either on a parabola, hyperbola, or ellipse. The curve degenerates to a straight line when the direction of motion is parallel to $L$ and when the initial position of $L$ is collinear with the origin. Also, we can show that the origin is on the same side of the collision front as one of the foci of the curve. Generally, the two endpoints of a collision front (accessible points) can be computed by setting $Y$ to $y_0$ and $y_0 + l$ in (1) and (2).

Having computed two endpoints of the collision front of a single movement, we must now compute all the endpoints of the collision fronts ($AVS$) when the robot is at a given start point with a given speed $v$. For a set of polygonal obstacles with $n$ vertices, there are $n$ candidate points. However, some points may not be accessible from the start point since some other movement intercepts the accessibility of that point. We now show that it takes $O(n \log n)$ time to compute $AVS$.

---

In general, the collision front is a portion of (i) a hyperbola, if $v_{\text{max}} > v_L \cos \theta$, (ii) a parabola, if $v_{\text{max}} = v_L \cos \theta$, (iii) an ellipse, if $v_{\text{max}} < v_L \cos \theta$. In our problem where $v_{\text{max}} > v_L > v_L \cos \theta$, the collision front is always a portion of a hyperbola except for degenerate cases.
The set of accessible points is determined by using a similar technique to that used to compute the visibility of a given set of line segments. In the following we show two methods for finding elements of \( \text{AVS} \).

**Method 1 (Plane-Sweep).** This is based on a plane-sweep algorithm [Pr]. This is a two-step process. The first pass sorts all the vertices, say in a clockwise direction with respect to \( O \). This sorting process takes \( O(n \log n) \) time. The second pass rotates a line about \( O \). It halts each time the line intersects a vertex, and checks whether or not the collision front associated with the vertex is accessible from \( O \). This process can be achieved in \( O(\log n) \) time by using a 2-3 tree [Ah] to maintain the active collision fronts based on their distance from \( O \). In this way the closest collision front is marked as accessible from \( O \). Since it takes at most \( O(n \log n) \) time to build the initial 2-3 tree and \( O(\log n) \) time for each update at a vertex, the determination of accessibility of \( n \) candidate points takes \( O(n \log n) \) time. Therefore, given a start point and a set of candidate points, it takes \( O(n \log n) \) time to compute accessible vertices from the start point. Alternately, since no two collision fronts intersect, we can use the approach of Asano et al. [As] to find the accessible points in \( O(n) \) time after the sorting pass. This gives no asymptotic speedup, but it does avoid the use of 2-3 trees.

**Method 2 (Merge Sort).**\(^9\) In this method the sorting and closest-front finding passes are combined into a single process. The set of collision fronts is divided into two equal-sized groups. For each group we find recursively a polar-order list that describes the collision fronts visible from \( O \). Each list element corresponds to an angular interval in which a particular collision front is closest to \( O \). The list for the whole set of collision fronts is found by merging the lists for the two half-sets. The endpoints of the angular intervals in the two lists determine a finer set of intervals. These finer intervals are computed during a walk through the two lists in tandem; in each of the new intervals we determine which collision front is closer to \( O \). Finally, we merge adjacent intervals in which the closest collision front is the same to produce the list for the whole set. Since it takes \( O(n) \) time to combine the two lists, the entire algorithm for finding \( \text{AVS} \) takes \( O(n \log n) \) time.

At this point note that, as a result of our theorem, a motion that meets with the same vertex of an obstacle more than once is not time-minimal. As there are \( n \) vertices in the environment, an \( \text{AVS} \) needs to be generated at most \( n \) times before a time-minimal motion reaches the destination point. This observation leads to an asymptotic computation time of \( O(n^2 \log n) \) to compute a motion using procedure PlanMotion.

Figure 10 shows a scene with 16 moving obstacles. Points \( O \) and \( G \) are the start and destination points, respectively. Each polygonal obstacle moves in the direction indicated by the arrow near the obstacle at a velocity given in parentheses. In Figures 11(a)–(d), the thick arrows show the motion generated by our algorithm.

\(^9\)This alternative algorithm for finding \( \text{AVS} \) is suggested by a referee.
Hatched obstacles indicate the locations of the obstacles when the point robot is at the end of the thick arrow, and dotted obstacles show the locations of the obstacles from the previous figure. For an environment containing this number of obstacles, it takes about 0.5 seconds to compute a motion using a VAX 11/785.

Note that for $m$ stationary convex polygonal obstacles with a total of $n$ vertices, a shortest length path can be computed using the visibility graph in $O(n^2)$ time [As], [We], $O(E + n \log n)$ time [Gh], or $O(E_s + n \log n)$ time [Kap], where $E$ is the number of edges in the visibility graph and $E_s$ is the number of edges in the visibility graph that are locally tangent at both endpoints. The quantities $E$ and $E_s$ are bounded by $n^2$ and $m^2$, respectively.

5. Concluding Remarks. We have studied the problem of moving a point robot among a set of obstacles moving at constant velocities in the plane. We have discussed the case where the robot can move faster than the obstacles and the destination point. Making use of the concept of accessibility, we have demonstrated an $O(n^2 \log n)$ algorithm to find a motion, and proved that the motion found is time-minimal.

Our main result required that the speed of the destination point be less than
Fig. 11. A time-minimal motion: (a) $t = 2.5$, (b) $t = 6.1$, (c) $t = 7.1$, and (d) $t = 14.4$. 
that of the robot. This restriction is not necessary. Of course, when the destination point moves faster than the robot, it is possible that the robot will not be able to reach the destination point. This can be resolved by applying procedure \textit{Plan Motion}.\footnote{A point that moves faster than \(n_{\text{max}}\) may be accessible either at zero, one, or two locations. If the destination point is accessible at two locations, then the one with a younger accessible time is adopted at its accessible point.} If all the vertices have been visited before finding the destination point, then \textit{PlanMotion} reports that it is not possible to reach the destination point. To conclude correctly that the destination point is unreachable, we must modify the proof of the base case of the theorem. The proof of the inductive case needs no modification since we did not have to use this restriction.

We now show how to relax our original assumption that the number of steps in a motion of the destination point is bounded. Let \(n_{\text{goal}}\) be the number of steps made by the destination point. Earlier we required it to be bounded so that the accessible point of the destination point could be determined in constant time. This can be done in \(O(n_{\text{goal}})\) time, so it does not affect the total computation time of \(O(n^2 \log n)\) as long as \(n_{\text{goal}} = O(n)\).

\textbf{Acknowledgments.} We have benefited greatly from the incisive critical comments of an anonymous referee. We would also like to thank Azriel Rosenfeld for his comments.

\textbf{References}


Planning a Time-Minimal Motion Among Moving Obstacles


