

SAND99-0458C

# Shortest Path Planning for a Tethered Robot or an Anchored Cable\*

RECEIVED

MAR 03 1999

OSTI

Patrick G. Xavier  
 Sandia National Laboratories  
 Albuquerque, NM 87185-1008

## Abstract

We consider the problem of planning shortest paths for a tethered robot with a finite length tether in a 2D environment with polygonal obstacles. We present an algorithm that runs in time  $O((k_l+1)^2 n^4)$  and finds the shortest path or correctly determines that none exists that obeys the constraints; here  $n$  is the number obstacle vertices, and  $k_l$  is the number loops in the initial configuration of the tether. The robot may cross its tether but nothing can cross obstacles, which cause the tether to bend. The algorithm applies as well for planning a shortest path for the free end of an anchored cable.

## 1 Introduction

### 1.1 Overview

While motion planning has been one of the more intensely studied areas of robotics, motion planning for tethered robots among obstacles has been relatively unexplored. This paper considers the problem of planning the shortest path for a planar mobile robot with a finite, anchored tether. In this problem, the robot must move from a given position and given tether configuration to a goal position and arbitrary tether configuration without violating the problem constraints.

We provide an  $O((k_l+1)^2 n^4)$  algorithm, where  $n$  is the geometric complexity of the planar environment and  $k_l$  is the number of loops around obstacles in the tether's initial configuration. Although an infinitely flexible, automatically retracting tether is assumed, we believe that our re-

sults will have applicability in motion planning when the tether has bounded curvature and in motion planning for serpentine, or hyper-redundant, robots. A more precise complexity analysis is given in the body of the paper.

### 1.2 Previous and Related Work

A great variety of motion planning problems have been studied extensively; for summaries, see Latombe [16] and Halperin, et al [8]. While the only provably-good polynomial-time algorithms for planning *time-optimal* obstacle-avoiding motions are approximation algorithms (e.g., [6]), exact and approximate polynomial-time algorithms exist for a wide range of *shortest-path* problems; see, Mitchell [17] for a summary. However, shortest path planning for tethered robots appears to be an unexplored domain.

Prior work on tethered robots has concentrated on generating motions for multiple robots that do not result in their tethers getting tangled. Sinden [19] presents an algorithm for scheduling the motion of multiple planar (point) robots that each have an umbilical (rigid tether) that must not cross the umbilicals of other robots. Pardo-Castellote and Martins [18] consider a related run-time problem in which the umbilicals are perpendicular to the boundary of the environment, the anchor points move on this boundary, the robots are restricted to rectilinear paths, and there may be non-robot obstacles.

Hert and Lumelsky considered planning trajectories for planar robots with flexible tethers of finite length that cannot be crossed but instead are deformed by robots pushing them. Their first result [11] is a general exhaustive-search algorithm for finding an optimal (e.g., minimum total cable length) robots-and-tether configuration in which the robots reach their goals. They also developed [12] an  $O(n^4)$  algorithm for planning an efficient set of robot tra-

\*This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy.

## **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## **DISCLAIMER**

**Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.**

jectories that results in a given target configuration of tethers, assuming that each robot begins at the anchor point of its tether. An extension [13] plans trajectories such that each robot  $i$  visits a given series of  $p$  target points  $\{T_i^p\}$ . Excluding computation of the sequence of tether configurations, this algorithm has complexity  $O(n^4 \log n)$ . More recently, they developed an algorithm for planning tangle-free motions [14] in three dimensions.

It is intuitive that finding a shortest path for a tethered robot is similar to solving the same problem for the “head” of a corresponding serpentine manipulator whose length, shape, and curvature limits match those of the combined robot and tether. However, generating a configuration-space (joint-space) path that corresponds to a serpentine robot “slithering” along a path in world space has been a research topic itself. This was the focus of Chirikjian and Burdick’s work on obstacle avoidance in [2], which assumed that a set of safe “tunnels” had already been found in the workspace. More recently, Choset and Burdick have investigated sensor-based planning methods using Generalized Voronoi Graph constructions [4, 5]. Henning, Hickman, and Choset then developed an algorithm [9] that combines these methods and others [3] to perform motion planning for serpentine robots. The algorithm is fast, but it is not complete for robots of finite length, even if they have no curvature limits.

## 2 Results

### 2.1 Problem Formulation

In an instance  $(\mathcal{O}, L, \mathbf{X}, \mathbf{u}, \mathbf{s}, \mathbf{g}, c)$  of the shortest path planning problem for a tethered robot with finite-length tether, the robot is a point that moves in a planar environment  $\mathcal{O}$  of disjoint simple polygons with  $n$  total vertices. This point robot is attached to an anchor point  $\mathbf{u}$  in the plane by a tether that has maximum length  $L$ . The tether’s initial configuration  $\mathbf{X}$  is given by a path no longer than  $L$  from  $\mathbf{u}$  to the start position  $\mathbf{s}$  of the robot.  $\mathbf{X}$  does not intersect the interior of any obstacles. For simplicity, we will assume that the tether automatically retracts so that it is kept taut; thus,  $\mathbf{X}$  is a sequence of line segments.  $\mathbf{g}$  is the goal position the robot must move to.

$c$  specifies the type of constraints a solution path must obey. In its motion, the robot must not enter any of the obstacles. The tether slides freely in obstacle-free regions and on obstacles. It is physically prevented from entering the obstacles, which cause it to bend to maintain this

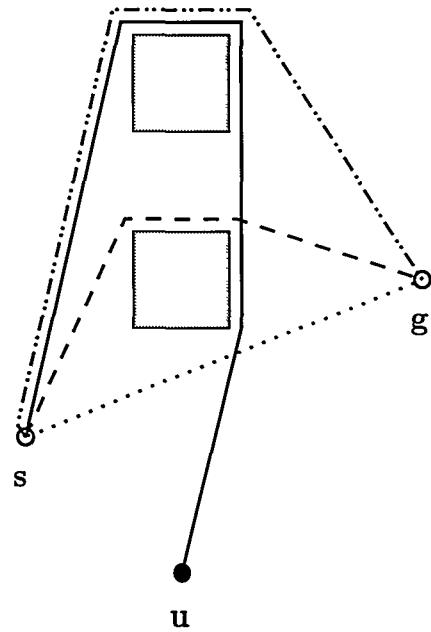


Figure 1: Robot is at  $\mathbf{s}$ ; solid line shows tether in initial configuration, with anchor point  $\mathbf{u}$ . Robot must go to  $\mathbf{g}$ . Dotted line is shortest path without tether constraint. Dashed line is shortest path obeying semi-planar crossing constraint. Fancy line is shortest path obeying purely planar crossing constraint.

condition. The tether length must never exceed  $L$ . These constraints are called the *tether and obstacle constraints*. If the robot can cross the tether, we say the robot is *semi-planar*; if it cannot, then the robot is *purely planar*; and if in addition, it can push the tether to avoid crossing the tether, it is a *pushy planar* robot. These constraints are called the *crossing constraints*. For the given crossing constraint  $c$ , a solution to the problem instance is a shortest path from  $\mathbf{s}$  to  $\mathbf{g}$  such that a robot that starts with the tether in configuration  $\mathbf{X}$  and follows the path obeys the tether and obstacle constraints and the crossing constraint. We say that such a path *obeys the tether, obstacle, and crossing constraints under the initial conditions*.

Figure 1 illustrates a problem and optimal solutions under semi-planar and purely planar crossing constraints.

### 2.2 Statement of Results

Our main result is an algorithm that solves the shortest path planning problem for a tethered robot with a finite-length tether under the semi-planar crossing constraint. We introduce geometric and topological characteristics of the initial configuration as additional measures of problem

complexity. We now give our main theorem, referring to our algorithm as ‘‘Algorithm SP’’.

**Theorem 2.1** *If  $(\mathcal{O}, L, \mathbf{X}, \mathbf{u}, \mathbf{s}, \mathbf{g}, \text{semi-planar})$  is an instance of the shortest path planning problem for a tethered robot with finite-length tether, then Algorithm SP will return a minimal-length solution path from  $\mathbf{s}$  to  $\mathbf{g}$  that obeys the constraints under the initial conditions, if such a solution exists. Otherwise, the algorithm will correctly determine that there is no solution that satisfies the constraints. In both cases, if  $\mathbf{X}$  has  $k_s$  straight sections and  $k_l$  loops, and  $\mathcal{O}$  has  $n$  vertices, then the algorithm will run in time  $O((k_l + 1)k_s n^3)$ .*

Since the number of straight sections will be no greater than the number of vertices multiplied by one more than the number of loops,  $k_s \leq (k_l + 1)n$ . This yields the following corollary.

**Corollary 2.2** *If  $(\mathcal{O}, L, \mathbf{X}, \mathbf{u}, \mathbf{s}, \mathbf{g}, \text{semi-planar})$  is an instance of the shortest path planning problem for a tethered robot with finite-length tether, then Algorithm SP will return a minimal length solution path from  $\mathbf{s}$  to  $\mathbf{g}$  that obeys the constraints under the initial conditions, if such a solution exists. Otherwise, the algorithm will correctly determine that there is no solution that satisfies the constraints. In both cases, if  $\mathbf{X}$  has  $k_l$  loops, and  $\mathcal{O}$  has  $n$  vertices, then the algorithm will run in time  $O((k_l + 1)^2 n^4)$ .*

Different, possibly more useful ways to express the complexity of the algorithm are discussed in Section 4.

### 3 Algorithms

#### 3.1 Overview and Preliminaries

We now describe our algorithm, show its correctness, and bound its complexity. We present the main algorithm in Section 3.2. We analyze its time complexity in Section 3.3, making much use of results from computational geometry. We show the algorithm’s correctness in Section 3.4. We first review some concepts necessary to understand our results.

A *pulled-taut configuration* of a tether results from retracting it from either end until taut, without the robot moving, bending only around obstacle vertices. Observe that the path following the tether in configuration  $\mathbf{X}$  from  $\mathbf{u}$  to the robot along its pulled-taut configuration  $\tilde{\mathbf{X}}$  is the shortest path in its homotopy class, where the obstacles

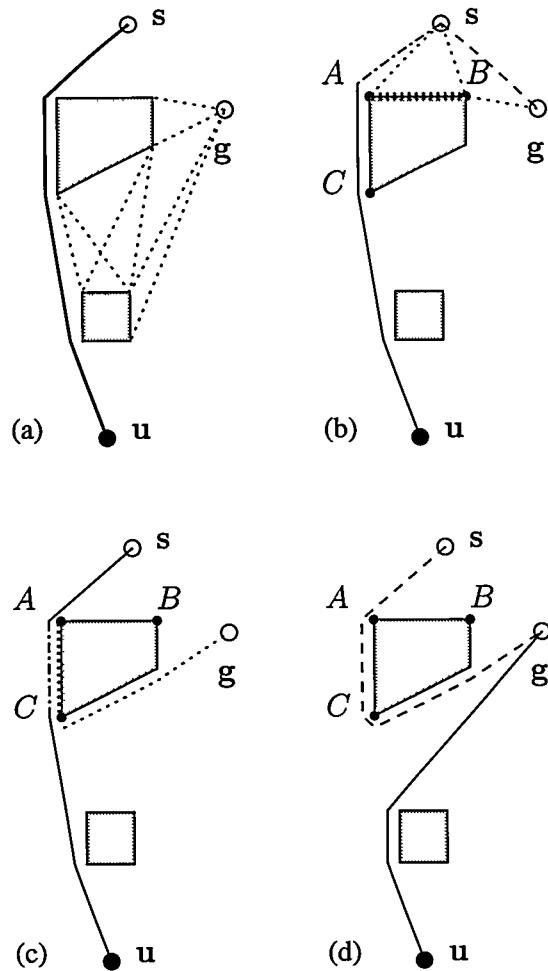


Figure 2: Algorithm SP. (a) Problem instance and visibility graph. (b) Path generated by considering shortest paths from vertices in set  $V_0 = \{A, B\}$  visible from  $S_0$ . (c) Further backtracking along tether finds vertex  $C$  newly visible from  $S_1$ ;  $V_1 = \{C\}$ . (d) Solution. (Paths and tether are shown offset from obstacles so they can be seen.)

are regarded as holes in the domain. We note that the minimum-length path of a homotopy type (i.e., within a homotopy class) is sometimes called a *taut string path*.

Let us be given a planar environment containing a polygonal set of planar obstacles  $\mathcal{O}$  with  $n$  vertices, and let this environment be triangulated such that all obstacle edges are triangle edges. If  $x$  is a point in the environment, let  $T(x)$  denote the triangle containing  $x$ . Let  $p : [0, l] \rightarrow \mathbb{R}^2$  be an obstacle-free path containing the minimal-length sequence  $\{x_i\}$  of  $m + 1$  vertices lying in the interiors of (non-obstacle) triangles in the triangulation such that (a)  $x_0 = p(0)$  and  $x_m = p(l)$ , (b) for all  $i$ ,

$T(x_i)$  is edge adjacent to  $T(x_{i+1})$ , and (c) for all  $i$ , there is no point on the segment of  $p$  from  $x_i$  to  $x_{i+1}$  that does not lie in  $T(x_i) \cup T(x_{i+1})$ . We say that the sequence  $\{x_i\}$  describes path  $p$  in the triangulation.

### 3.2 The Main Algorithm

We present a simple algorithm that takes as input an instance  $(\mathcal{O}, L, \mathbf{X}, \mathbf{u}, \mathbf{s}, \mathbf{g}, \text{semi-planar})$  of the shortest path planning problem for a tethered robot with finite-length tether. The algorithm will be given at a fairly high level, using well-known or easy computational geometry algorithms as sub-algorithms. We give additional detail in the complexity analysis in Section 3.3.

#### Algorithm SP

Let  $p_{\mathbf{X}} : [0, l] \rightarrow \mathbb{R}^2$  denote a path from  $\mathbf{u}$  to  $\mathbf{s}$  that follows  $\mathbf{X}$ , where  $l$  is the length of the tether in configuration  $\mathbf{X}$ . Then define the reverse path  $r_{\mathbf{X}}$ , where  $r_{\mathbf{X}}(w) = p_{\mathbf{X}}(l - w)$  for all  $w \in [0, l]$ . We use these functions and their restrictions to sub-ranges of  $[0, l]$  only to avoid geometric ambiguity in the description; implementation does not require actually constructing and manipulating these functions.

1. Triangulate the environment such that all obstacle edges are triangle edges, and label each triangle as being an obstacle triangle or a freespace triangle.
2. Compute the visibility graph  $\mathcal{G}_v$  of  $\mathcal{O} \cup \{\mathbf{g}\}$ , and use it to compute the shortest path from each vertex to  $\mathbf{g}$ .
3. Find the vertices of  $\mathcal{O} \cup \{\mathbf{g}\}$  visible from  $\mathbf{s}$  and order them by angle relative to  $\mathbf{s}$ .
4. Compute the changes that occur in the ordered set of vertices visible from  $r_{\mathbf{X}}(w)$  as  $w$  goes from 0 to  $l$ , and compute the ordered set of sections  $\{S_i\}$  of  $r_{\mathbf{X}}$  on which this ordered set does not change. (Some sections may have zero length if general position is not assumed.)
5. For each  $S_i$ , with corresponding interval  $[w_i, w_{i+1}]$  such that  $r_{\mathbf{X}} : [w_i, w_{i+1}] \mapsto S_i$ , in order:
  - (a) For some point  $x_i \in S_i$  and corresponding  $x_i \in [w_i, w_{i+1}]$  such that  $r_{\mathbf{X}}(x_i) = x_i$ , compute the set of vertices  $\mathcal{V}_i$  containing
    - i. each visible vertex if  $i = 0$ ;
    - ii. each visible vertex that was not visible in the previous section, otherwise.

(b) For each vertex  $v$  in  $\mathcal{V}_i$ :

- i. Let path  $q_{i,v}$  be the path that is the concatenation of  $p_{\mathbf{X}}|[0, l - x_i]$ , the line segment from  $x_i$  to  $v$ , and the shortest path from  $v$  to  $\mathbf{g}$ .  $(p_{\mathbf{X}}|[0, l - x_i])$  denotes “ $p_{\mathbf{X}}$  restricted to  $[0, l - x_i]$ ”.
- ii. Compute the minimum length path  $\tilde{q}_{i,v}$  that is homotopy equivalent to  $q_{i,v}$ .
- iii. If the length of  $\tilde{q}_{i,v}$  is greater than  $L$ , then skip the rest of this loop body, go to step 5(b)i, and consider the next vertex in  $\mathcal{V}_i$ .
- iv. (Otherwise), let path  $p_{i,v}$  be the path that is the concatenation of  $r_{\mathbf{X}}|[0, x_i]$ , the line segment from  $x_i$  to  $v$ , and the shortest path from  $v$  to  $\mathbf{g}$ .
- v. Compute the minimum length path  $\tilde{p}_{i,v}$  that is homotopy equivalent to  $p_{i,v}$ , compare the length of  $\tilde{p}_{i,v}$  to that of best solution so far, and set the best solution to  $\tilde{p}_{i,v}$  if its length is shorter.

### 3.3 Complexity

We now give a lemma on the complexity of Algorithm SP.

**Lemma 3.1** *Let  $(\mathcal{O}, L, \mathbf{X}, \mathbf{u}, \mathbf{s}, \mathbf{g}, \text{semi-planar})$  be an instance of the semi-planar tethered robot problem with bounded tether and static obstacles, with  $\mathbf{X}$  being a taut configuration. Let  $k_s$  be the number of straight sections of  $\mathbf{X}$ , and let  $k_l$  be the number of loops in  $\mathbf{X}$ . Then the Algorithm SP will run in time  $O((k_l + 1)k_s n^3)$  on the given problem instance.*

**Proof:** We go through the steps of the algorithm.

Step 1 can be done in time  $O(n \log n)$  using the method of Hertel and Mehlhorn [15].

In Step 2, we use Ghosh and Mount’s algorithm [7] to construct the visibility graph and Dijkstra’s algorithm to find the set of shortest paths. Both algorithms take time  $O(n \log n + E)$ , in time, where  $E$  is the number of visibility edges, which is  $O(n^2)$  but often less in practice. Step 3 requires time  $O(n \log n)$ , since there are at most  $O(n)$  visible vertices, and the visibility graph was previously computed.

In step 4, there are  $O(n^2)$  changes in the ordered set of visible vertices for each straight segment of  $\mathbf{X}$ , and computing these changes costs  $O(n^2 \log n)$  per segment.

This can be done by considering each vertex and computing the regions of the segment that are visible from it. Using a rotational sweep algorithm, this can be done in time  $O(n \log n)$  for each vertex: we sort the edge vertices in angular order about the viewing vertex, and then we rotationally sweep a ray from this vertex, keeping track of which edges intersect it between the vertex and its intersection with the segment. Doing this for all vertices in  $\mathcal{O} \cup \{g\}$  takes time  $O(n^2 \log n)$ , as does ordering the changes in the set of vertices visible from a point traversing the segment. Therefore, this step takes time  $O(k_s n^2 \log n)$  and finds  $O(k_s n^2)$  visibility changes. This analysis also covers the cost of step 5(a).

Finally 5(b), since  $q_{i,v}$  and  $p_{i,v}$  will each be described by  $O((k_l + 1)n)$  points, computing the minimum-length homotopy equivalent paths  $\tilde{q}_{i,v}$  and  $\tilde{p}_{i,v}$  will take time  $(O(k_l + 1)n)$  using Hershberger's and Snoeyink's algorithm [10]. Therefore, the total cost of the step 5 is  $O((k_l + 1)k_s n^3)$ , which is greater than that of any of the other steps. This gives us the final complexity bound.  $\square$

### 3.4 Correctness

The following lemma states algorithm correctness.

**Lemma 3.2** *Let  $(\mathcal{O}, L, \mathbf{X}, \mathbf{u}, \mathbf{s}, \mathbf{g}, \text{semi-planar})$  be an instance of the semi-planar tethered robot problem with bounded tether and static obstacles, with  $\mathbf{X}$  being a taut configuration. Then, given this input, Algorithm SP will find a shortest path solution that obeys the problem constraints if such a solution exists.*

**Proof:** We show that if  $\mathbf{p}_{min}$  is a minimum length robot path from  $\mathbf{s}$  to  $\mathbf{g}$  that obeys the constraints, then the algorithm will find an equivalently good solution. (Figures 3 and 4 should be helpful.)

Let  $\mathbf{p}_X : [0, l] \rightarrow \mathbb{R}^2$  be the path from  $\mathbf{u}$  to  $\mathbf{s}$  that follows  $\mathbf{X}$ , where  $l$  is the length of the tether in configuration  $\mathbf{X}$ . Let  $l_{sol}$  be the length of  $\mathbf{p}_{min}$ . Let  $l_q = l + l_{sol}$ , and define  $\mathbf{q}_{min} : [0, l_q] \rightarrow \mathbb{R}^2$  be  $\mathbf{p}_X * \mathbf{p}_{min}$ , the concatenation of  $\mathbf{p}_X$  and  $\mathbf{p}_{min}$ . Let  $\tilde{\mathbf{q}}_{min} : [0, l_{\tilde{q}}] \rightarrow \mathbb{R}^2$  be the minimum-length path in  $\mathbf{q}_{min}$ 's homotopy class, with length  $l_{\tilde{q}}$ . Note that  $\mathbf{p}_X$  and  $\mathbf{p}_{min}$  are minimum length in their homotopy classes.

Because  $\mathbf{p}_X$ ,  $\mathbf{p}_{min}$ , and  $\tilde{\mathbf{q}}_{min}$  can only bend around obstacle vertices and because  $\mathbf{q}_{min}$  is homotopy equivalent to  $\tilde{\mathbf{q}}_{min}$ , we can consider  $\mathbf{q}_{min}$  in five sections. (Any of these may have length zero.) For some  $x_g \in [0, l_q]$ ,

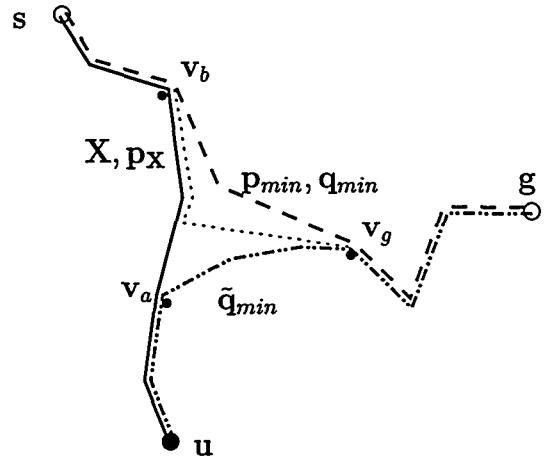


Figure 3: Paths and vertices described in proof.  $\mathbf{p}_{min}$  is solution.  $\mathbf{q}_{min} = \mathbf{p}_X * \mathbf{p}_{min}$ , so that these shortest path  $\tilde{\mathbf{q}}_{min}$  in its homotopy class is the pulled taut configuration of the tether after the robot follows path  $\mathbf{p}_{min}$ . Because of homotopy equivalence and because  $\mathbf{p}_X$ ,  $\mathbf{p}_{min}$ , and  $\tilde{\mathbf{q}}_{min}$  can only bend around obstacle vertices,  $v_g$  must be visible from  $\mathbf{p}_X$  with respect to  $\mathbf{p}_{min}$  and  $\tilde{\mathbf{q}}_{min}$ . It follows that the algorithm will find  $\mathbf{p}_{min}$ .

the terminal segment  $\mathbf{q}_{min}|[x_g, l_q]$  of  $\mathbf{q}_{min}$  beginning at  $v_g = \mathbf{q}_{min}(x_g)$  and ending at  $\mathbf{g}$  must be coincident with a terminal segment  $\tilde{\mathbf{q}}_{min}|[y_g, l_{\tilde{q}}]$  of  $\tilde{\mathbf{q}}_{min}$  such that  $\tilde{\mathbf{q}}_{min}(y_g) = v_g$ . Similarly, for some  $x_a$  and  $y_a$ , the initial segment  $\mathbf{q}_{min}|[0, x_a]$  of  $\mathbf{q}_{min}$  beginning at  $\mathbf{s}$  and ending at  $v_a = \mathbf{q}_{min}(x_a)$  must be coincident with an initial segment  $\tilde{\mathbf{q}}_{min}|[0, y_a]$  of  $\tilde{\mathbf{q}}_{min}$  and identical to an initial segment of  $\mathbf{p}_X$ . There may be three additional segments of interest. In the first,  $\mathbf{q}_{min}|[x_a, l]$ ,  $\mathbf{q}_{min}$  is identical to  $\mathbf{p}_X$  but not coincident with any segment of  $\tilde{\mathbf{q}}_{min}$ . In the second,  $\mathbf{q}_{min}|[l, x_b]$  for some  $x_b \leq x_g$ ,  $\mathbf{q}_{min}$  backtracks along  $\mathbf{p}_X$ . If  $x_b < x_g$ , then in the third segment  $\mathbf{q}_{min}|[x_b, x_g]$ ,  $\mathbf{q}_{min}$  is not backtracking over  $\mathbf{p}_X$  and is not coincident with any terminal section of  $\tilde{\mathbf{q}}_{min}$ .

Because  $\mathbf{q}_{min}$  and  $\tilde{\mathbf{q}}_{min}$  are homotopy equivalent and bend only around obstacle vertices except at  $\mathbf{s}$ , the path segments  $\mathbf{q}_{min}|[x_b, x_g]$  and  $\tilde{\mathbf{q}}_{min}|[y_a, y_g]$ , must each bend in only one direction, bend away from each other, and not cross. Furthermore, it follows that there is some section of  $\mathbf{p}_X$  from which  $v_g$  is visible with respect to  $\mathbf{q}_{min}|[x_b, x_g]$ , and  $\tilde{\mathbf{q}}_{min}|[y_a, y_g]$ . A path that is a concatenation of  $\mathbf{q}_{min}|[l, x_b]$  and the path segment backtracking over  $\mathbf{p}_X$  to a point in this section and proceeding via the visibility line segment to  $v_g$  is homotopy equivalent to  $\mathbf{p}_{min}$ . It follows that the algorithm will find  $\mathbf{p}_{min}$  and

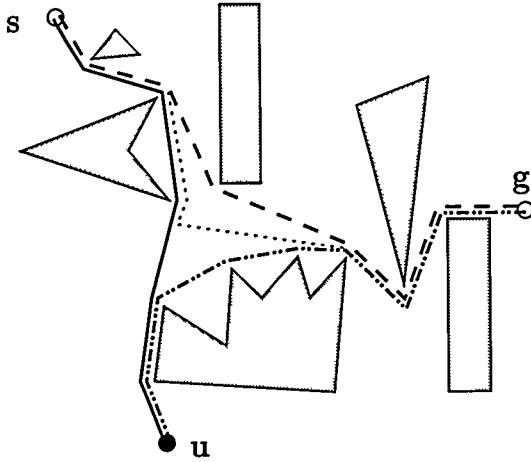


Figure 4: Paths from the proof, shown with obstacles.

return it if an equivalently good path was not found first.

Finally, we show that for all  $x \in [0, l_q]$ , the minimum-length path that is homotopy equivalent to  $q_{min}|[0, x]$  has length no greater than  $L$ . Let  $v_b = q_{min}(x_b)$ . Because  $q_{min}$  and  $\tilde{q}_{min}$  are homotopy equivalent, the minimum-length path that is homotopy equivalent to  $q_{min}|[0, x]$  has length no greater than  $L$  for  $x \notin [x_b, x_g]$ . To see that this condition is also true for  $x \in [x_b, x_g]$ , we observe that because  $q_{min}$  only bends around obstacle vertices and  $q_{min}$  is homotopy equivalent to  $\tilde{q}_{min}$ ,  $q_{min}|[x_b, x_g]$  never leaves the triangle  $v_a v_b v_g$ . Since  $q_{min}$  is homotopy equivalent to  $\tilde{q}_{min}$  and the minimum-length paths homotopy equivalent to  $q_{min}|[0, x_b]$  and  $q_{min}|[0, x_g]$  have length no greater than  $L$ , this implies that the minimum-length path that is homotopy equivalent to  $q_{min}|[0, x]$  has length no greater than  $L$  for  $x \in [x_b, x_g]$ .  $\square$

Theorem 2.1 follows from Lemmas 3.1 and 3.2. ( $\square$ )

## 4 Future Work Discussion

### 4.1 Purely Planar and Pushy Planar Crossing Constraints

We expect Algorithm SP can be adapted to solve the version of the shortest path planning problem for a tethered robot with a finite tether and pure planar constraints and the version with pushy planar constraints. In the most direct extension, for pure planar constraints, we would treat

the segments of  $\mathbf{X}$  as obstacle edges. As the search backtracks along the initial configuration  $\mathbf{X}$ , not only would the set of visible vertices change, but so would the edges of the visibility graph. This implies that the shortest paths from visible vertices to the goal could potentially be changed. However, we conjecture that the shortest path solution would not go through any vertex from which the shortest path to goal would change. If this is true, the only modifications to the current algorithm would be to rule out candidate paths that cross  $\mathbf{X}$  and to treat the anchor point  $u$  as an obstacle vertex. Coping with pushy planar crossing constraints poses additional problems. The most obvious of these is that a new sub-algorithm to compute pulled-taut configurations would be needed.

### 4.2 Observations and Conjectures

It has been noted that the number  $E$  of edges in the visibility graph is often much smaller than  $O(n^2)$ . Similarly, when moving from  $s$  to  $u$  along  $\mathbf{X}$  the number  $V_c$  of changes to the (ordered) set of visible vertices might be much smaller than  $O(k_s n^2)$ . In any case, we can write the complexity as  $O((k_l + 1)nV_c + E)$ . Furthermore,  $k_l n$  is a loose bound on the number of triangles visited by  $\mathbf{X}$ . If  $T_{\mathbf{X}}$  is the actual bound, then the complexity obeys an  $O((T_{\mathbf{X}} + n)V_c + E)$  or  $O((T_{\mathbf{X}} + n)k_s n^2 + E)$  bound.

We conjecture that the cost of computing the shortest path homotopy equivalent to  $p_{i,v}$  is  $O(n)$ , not  $O(T_{\mathbf{X}} + n)$ , because if tautness is imposed starting from the  $g$  end, then computation after the first loop is completed will be unnecessary. Computing each  $\tilde{q}_{i,v}$  individually appears to be  $O(T_{\mathbf{X}} + n)$ . However, it might be possible that after the first  $\tilde{q}_{i,v}$  is computed, we might be able to compute successive ones in  $O(n)$  or amortized  $O(n)$  time. This would result in a  $O(nV_c + T_{\mathbf{X}} + E)$  or  $O(k_s n^3 + T_{\mathbf{X}} + E)$  time bound. Finally, a radically different algorithm might do some sort of binary search along  $\mathbf{X}$  when considering sets of visible vertices. Although such an algorithm might potentially achieve a time bound of  $O((T_{\mathbf{X}} + n)n \log n)$ , work remains to be done to prove such an algorithm correct.

An efficient shortest path algorithm for tethered robots that have tethers with bounded curvature would be of substantial practical interest because a greater variety of cables would fit the model. We believe that this problem might be easier than the bounded curvature shortest path problem for a point robot (without a tether), for which no exact polynomial-time algorithm exists yet. The key

observation is that while the curvature of the tether is bounded, the curvature of the path is not.

## 5 Conclusions

We have presented an algorithm for solving the shortest path planning problem for a tethered robot with a finite length tether with semi-planar crossing constraints. We gave precise formulations for this problem and two closely related ones. Given an instance  $(\mathcal{O}, L, \mathbf{X}, \mathbf{u}, \mathbf{s}, \mathbf{g}, \text{semi-planar})$  of the shortest path planning problem for a tethered robot with finite-length tether, our algorithm will return a minimal-length solution path from  $\mathbf{s}$  to  $\mathbf{g}$  that respects the problem constraints if such a solution exists, or it will correctly determine that there is no solution. The algorithm runs in time  $O((k_l + 1)k_s n^3)$ , where  $n$  is the number of vertices in  $\mathcal{O}$ ,  $k_l$  is the number of loops in initial cable configuration  $\mathbf{X}$ ,  $k_s$  is the number of straight sections of  $\mathbf{X}$ .

We also presented discussion about possible algorithms for solving the problem under the purely planar and pushy planar crossing constraints. We made observations about what the complexity would be in practice. In this setting, we obtain the bound  $O((T_{\mathbf{X}} + n)V_c + E)$ , where  $E$  is the number of edges in the visibility graph of  $\mathcal{O}$ ,  $T_{\mathbf{X}}$  the number of triangles visited by the  $\mathbf{X}$  in the triangulation of the environment, and  $V_c$  is the number of changes to the set of vertices of  $\mathcal{O}$  visible from a point moving along  $\mathbf{X}$  from  $\mathbf{s}$  to anchor point  $\mathbf{u}$ . Finally, we speculated on techniques that would lower the running time, and on the possibility of an algorithm for tethered robots whose tethers have bounded curvature.

**Acknowledgements** The author would like to thank Russell Brown, Tyson Henry, and Stephen Kaufman for their suggestions and comments at various stages of this work.

## References

- [1] J. Canny, B. Donald, J. Reif, and P. Xavier. On the complexity of kinodynamic planning. In *Proc. of the 29<sup>th</sup> Annual Symp. on the Foundations of Computer Science*, pages 306–316, White Plains, New York, 1988.
- [2] G. S. Chirikjian and J. W. Burdick. An obstacle avoidance algorithm for hyper-redundant manipulators. In *Proc. 1990 IEEE Int'l Conf. on Robotics and Automation*, pages 625–631, Cincinnati, OH, May 1990.
- [3] G. S. Chirikjian and J. W. Burdick. A modal approach to hyper-redundant manipulator kinematics. *IEEE Trans. on Robotics and Automation*, June 1994.
- [4] H. Choset and J. W. Burdick. Sensor-based planning, part i: The generalized voronoi graph. In *Proc. 1995 IEEE Int'l Conf. on Robotics and Automation*, Nagoya, Japan, 1995.
- [5] H. Choset and J. W. Burdick. Sensor-based planning, part ii: Incremental construction of the generalized voronoi graph. In *Proc. 1995 IEEE Int'l Conf. on Robotics and Automation*, Nagoya, Japan, 1995.
- [6] B. Donald, P. Xavier, J. Canny, and J. Reif. Kinodynamic motion planning. *Journal of the ACM*, 40(5), November 1993. Journal version of [1].
- [7] S. K. Ghosh and D. M. Mount. An output sensitive algorithm for computing visibility graphs. In *Proc. of the 27<sup>th</sup> Annual Symp. on the Foundations of Computer Science*, pages 11–19, 1987.
- [8] D. Halperin, L. Kavraki, and Latombe J.-C. Robotics. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 14, pages 755–778. CRC Press, 1997.
- [9] W. Henning, F. Hickman, and H. Choset. Motion planning for serpentine robots. In *ASCE Space and Robotics 98*, Albuquerque, NM, 1998.
- [10] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Computational Geometry: Theory and Applications*, 4:63–98, 1994.
- [11] S. Hert and V. Lumelsky. Computational geometry issues in the tethered robot problem. Tech. Report RL-94002, University of Wisconsin – Madison, Robotics Laboratory, February 1994.
- [12] S. Hert and V. Lumelsky. The ties that bind: Motion planning for multiple tethered robots. In *Proc. 1994 IEEE Int'l Conf. on Robotics and Automation*, pages 2734–2741, San Diego, CA, May 1994.
- [13] S. Hert and V. Lumelsky. Moving multiple tethered robots between arbitrary configurations. In *Proc. 1995 Int'l Conf. on Intelligent Robots and Systems*, August 1995.
- [14] S. Hert and V. Lumelsky. Motion planning in  $r^3$  for multiple tethered robots. In *Proc. 1997 Int'l Conf. on Intelligent Robots and Systems*, September 1997.
- [15] S. Hertel and K. Mehlhorn. Fast triangulation of the plane with respect to simple polygons. *Information and Control*, 64(1–3):52–76, 1985.
- [16] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [17] J. Mitchell. Shortest paths and networks. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 24, pages 445–466. CRC Press, 1997.
- [18] G. Pardo-Castellote and H. Martins. Real-time motion scheduling for a small workcell. In *Proc. 1991 IEEE Conf. on Robotics and Automation*, pages 810–817, April 1991.
- [19] F. W. Sinden. The tethered robot problem. *Int'l Journal of Robotics Research*, 9(1):122–133, February 1990.