OPTIMAL MOTION OF COVISIBLE POINTS AMONG OBSTACLES IN THE PLANE

(extended abstract)

Joseph S. B. Mitchell[*]
School of Operations Research
and Industrial Engineering
Cornell University
Ithaca NY 14853

Erik L. Wynters[†]
Center for Applied Mathematics
Cornell University
Ithaca NY 14853

Abstract

We consider the problem of determining an optimal pair of paths for two point robots that must remain covisible while moving in a plane cluttered with polygonal obstacles. We solve the (MIN-SUM) problem of minimizing the sum of the path lengths with an algorithm that requires time $O(E + n \log n)$ and space $O(E)$, where E is the size of the visibility graph induced by the set of obstacles. We also solve the (MIN-MAX) problem of minimizing the length of the longer path in time $O(E^2 + n^2 \log n)$ and space $O(E)$. In addition, we discuss the (MIN-TIME) problem of obtaining a coordinated motion parameterized by time that minimizes the time needed for both robots to reach their destinations. Assuming a common upper bound on velocity, we find a pair of paths together with a parameterization guaranteed to be within a factor of 2 of optimality.

1 Introduction

Imagine that two robots must communicate with one another while traveling among obstacles. The robot geometry, the obstacle geometry, and the effectiveness of communication can be modeled in various ways, but let's assume that we have point-sized robots, polygonal obstacles, and a communication system that works only when the robots can see one another. Assume, also, that initial and final positions for each robot have been specified, and that both robots have the same maximum speed. Under these assumptions, we consider the following optimal motion planning problems:

MIN-SUM Find a pair of paths between the initial and final positions that minimizes the sum of the two path lengths subject to the communication constraint.

MIN-MAX Find a pair of paths between the initial and final positions that minimizes the larger of the two path lengths subject to the communication constraint.

MIN-TIME Find a coordinated motion parameterized by time that minimizes the time needed for both robots to reach their destinations while maintaing communication.

Alternatively, we can view these problems as motion planning problems for a single robot with four degrees of freedom. In this case, imagine that the robot is a collapsible rod that can simultaneously translate, rotate, and expand or contract. We seek optimal motions for the rod, using the distances traveled by its endpoints as optimization criteria.

Prior work in this field suggested that shortest path planning for a robot with more than two degrees of freedom was inherently difficult. While shortest paths among polygonal obstacles for a point robot or a translating polygon (two degrees of freedom) can be found exactly in worst-case quadratic time, finding a shortest path for a point robot among polyhedra in three-space (three degrees of freedom) is NP-hard [CR]. The problem of moving a ladder (or any noncircular body free to rotate) in an optimal manner among obstacles in the plane (three degrees of freedom) is also very challenging and has been solved only for a restricted class of allowable motions [PS]. A characterization of optimal motions that doesn't presuppose a certain class of motions was obtained recently [IRWY], but it applies only when no obstacles are present.

We were, therefore, somewhat surprised to discover that we can solve the MIN-SUM problem, which admits *four* degrees of freedom and allows obstacles to be present, in the same complexity as the best shortest path algorithm for a *single* point robot with *two* degrees of freedom. Our algorithm requires time $O(E + n \log n)$ and space $O(E)$.

The main ideas behind our algorithm are as follows: we characterize the geometric form of optimal solutions; we bound the number of candidate paths satisfying these geometric constraints; and we use the theory of funnel trees and graph searching to enumerate and evaluate the candidate paths in time and space proportional to the number of such candidates.

A variation of our algorithm solves the MIN-MAX problem in time $O(E^2 + n^2 \log n)$ and space $O(E)$. More generally, we can solve the "bicriteria" version of the problem: given an upper bound on the length of one agent's path, find a motion of both covisible agents that minimizes the length of the other agent's path.

[*]Partially supported by NSF Grants IRI-8710858 and ECSE-8857642 and by a grant from Hughes Research Laboratories

[†]Supported by a grant from Hughes Research Laboratories

The MIN-TIME problem with bounded robot velocity seems to be quite difficult. We show that the optimal-time motion may correspond to a pair of paths that are not even locally optimal with respect to distance. Although we have not yet solved this problem exactly, we give a method of obtaining an approximate solution guaranteed to be within a factor of 2 of optimality.

2 Definitions and Preliminaries

Consider a set $\mathcal{O}$ of disjoint, simple-polygon obstacles defined by a total of n vertices. We define *freespace*, $\mathcal{F}$, to be the set of all points in the plane that are not contained in the relative interior of any obstacle. Note that the boundaries of obstacles are included in freespace. A *configuration* is an ordered pair of points in freespace that are visible to one another in the sense that the line segment joining them lies within freespace. Such points are called *covisible*, and the line segment connecting them is called a *feasible* line segment.

A *path function* is a continuous function from the unit interval $[0,1]$ into freespace that determines the position of a moving point at time $\tau \in [0,1]$. The image of a path function, the subset of freespace traversed by the point, is called a *path*; and any path function with image p is called a *parameterization* of p. We denote the *length* of path p by $\mu(p)$. A *path-pair* is an ordered pair of paths (p_1, p_2) and is *feasible* if there exist parameterizations f_1 and f_2 of p_1 and p_2, respectively, such that for all times τ the points $f_1(\tau)$ and $f_2(\tau)$ are covisible.

Now we can formally state the MIN-SUM problem: given an obstacle set $\mathcal{O}$, an initial configuration (s_1, s_2), and a final configuration (t_1, t_2), find a feasible path-pair (p_1, p_2) between these configurations that minimizes the sum $\mu(p_1) + \mu(p_2)$.

3 Characterizing Optimal Motions

For a path-pair (p_1, p_2) from (s_1, s_2) to (t_1, t_2) to be feasible it must satisfy a topological condition. Loosely speaking, we say that the paths must lie in the same "channel" between the obstacles. Putting it another way, the closed path from s_1 to s_1 using path p_1, line segment $\overline{t_1 t_2}$, path p_2, and line segment $\overline{s_2 s_1}$ must not contain any obstacles. We call the closed path described above the *path-polygon* of the path-pair (p_1, p_2) and call the paths and line segments the *definining paths* and *defining line segments*, respectively, of the polygon.

We say that a path-polygon is *obstacle-free* if it can be continuously deformed in freespace to a single point. More precisely, if P is a path polygon and v is a vertex of P, then P is obstacle-free if the path P has a parameterization homotopic to the constant function $f(\tau) = v$. When the path-polygon of a path-pair is obstacle-free, we say that the path-pair is *topologically feasible*. The following lemma states that topological feasibility is a necessary condition for feasibility.

Lemma 1 *If a path-pair is feasible, then it must be topologically feasible.*

Lemma 1 establishes that we can limit our search for an optimal path-pair to the set of topologically feasible ones. But there is a continuous family of topologically feasible path-pairs for every homotopy class of paths from s_1 to t_1. Since we find optimal path-pairs for the MIN-SUM and MIN-MAX problems by enumerating and comparing feasible candidates, we must reduce the search space further to do this efficiently.

We make the search efficient by considering only those path-pairs that are locally optimal. A locally-optimal path for the single-point shortest path problem is a path that cannot be shortened by making local changes; it is a "taut-string" path, which is known to consist of visibility graph edges ([Le], [SS], [Mi]). Similarly, a path-pair is *locally optimal* if each individual path is a taut-string path.

If a path-pair (p_1, p_2) is both topologically feasible and locally optimal, the paths p_1 and p_2 must satisfy exactly one of the following three conditions:

1. the paths are disjoint;
2. the intersection of the paths is a polygonal chain whose endpoints are vertices of the visibility graph (this includes the case in which the intersection is a single vertex); or
3. the intersection of the paths is a single point that is not a vertex of the visibility graph.

These three cases lead to three types of path-polygons: "hourglasses", "funnelglasses", and "bowties", respectively. We define these terms below and refer the reader to [MW] for a brief discussion of their history.

An *hourglass between a feasible line segment* $\overline{s_1 s_2}$ *and another feasible segment* $\overline{t_1 t_2}$ is an obstacle-free polygon bounded by these segments and two *disjoint* paths p_1 and p_2 where p_1 is a locally optimal path from s_1 to t_1, and p_2 is a locally optimal path from s_2 to t_2.

A *funnel between vertex v and line segment* $\overline{s_1 s_2}$ is a simple, obstacle-free polygon bounded by $\overline{s_1 s_2}$, a locally optimal path from v to s_1 and a locally optimal path from v to s_2. The vertex v is called the *apex* of the funnel, and the segment $\overline{s_1 s_2}$ is called the *base*. The path from apex to base with the interior of the funnel on its left side is called the *lower chain* of the funnel and the other path is called the *upper chain*.

A *funnelglass between a feasible line segment* $\overline{s_1 s_2}$ *and another feasible segment* $\overline{t_1 t_2}$ is a pair of funnels (f_1 and f_2 with bases $\overline{s_1 s_2}$ and $\overline{t_1 t_2}$, respectively, joined by a shortest path between their apices in such a way that both the path p_1 from s_1 to t_1 and the path p_2 from s_2 to t_2 are taut.

A *bowtie between a feasible line segment* $\overline{s_1 s_2}$ *and another feasible segment* $\overline{t_1 t_2}$ is a polygon formed from these line segments and two locally optimal paths, p_1 from s_1 to t_1 and p_2 from s_2 to t_2, that intersect at a single point that is not a vertex of the visibility graph.

In summary, we can characterize fully the set of locally optimal, topologically feasible path pairs:

Lemma 2 *If a path-pair is locally optimal and topologically feasible, then its path-polygon must be an hourglass, a funnelglass, or a bowtie.*

Our next lemma states that locally optimal, topologically feasible path-pairs not only form characteristic shapes but also admit feasible parameterizations.

Lemma 3 *If a path-pair is locally optimal and topologically feasible, then it must be feasible.*

Lemma 3 implies that an optimal path-pair must be locally optimal. Otherwise, it could be shortened and the new version would still be feasible. This, together with Lemma 2, gives the following theorem.

Theorem 1 *An optimal path-pair for the MIN-SUM problem is obtained by selecting one for which the path-polygon is an hourglass, a funnelglass, or a bowtie, and the associated cost $(\mu(p_1) + \mu(p_2))$ is minimum.*

Theorem 1 establishes the correctness of our algorithm, which appears in section 4. The efficiency of our algorithm depends on the number of hourglasses, funnels, and bowties that can occur. In section 5, we show that the following combinatorial bound on the number of these objects holds:

Theorem 2 *There are at most $O(E)$ hourglasses, funnels, and bowties.*

4 The MIN-SUM Algorithm

Instance: A set $\mathcal{O}$ of polygonal obstacles, an initial configuration (s_1, s_2), and a final configuration (t_1, t_2).

Algorithm:

1. Compute the visibility graph $VG_1(\mathcal{O})$ for the obstacle set and the points s_1, s_2, t_1, and t_2 treating $\overline{s_1 s_2}$ as a line segment obstacle, and compute the visibility graph $VG_2(\mathcal{O})$ for the obstacle set and the points s_1, s_2, t_1, and t_2 treating $\overline{t_1 t_2}$ as a line segment obstacle.
2. Generate all funnels with base $\overline{s_1 s_2}$ or $\overline{t_1 t_2}$.
3. Generate all hourglasses and bowties from $\overline{s_1 s_2}$ to $\overline{t_1 t_2}$ and determine which has the best associated path-pair.
4. Define an *augmented visibility graph* $\mathcal{G}$ as follows:
    - Beginning with the visibility graph $VG_1(\mathcal{O})$ obtained in step 1, create an additional "funnel" node fn for each funnel f generated in step 2.
    - Connect each funnel-node fn to the node v corresponding to the apex of funnel f. Let the length of edge (fn, v) be half the sum of the two lengths of the defining paths of f;
    - Create a "supersource" node s, linking it with an edge of length 0 to each funnel-node corresponding to a funnel based on $\overline{s_1 s_2}$. Similarly, create and link a "supersink" node t to each funnel-node corresponding to a funnel based on $\overline{t_1 t_2}$.
5. Find a shortest path from s to t in the graph $\mathcal{G}$. This determines the best path-pair for which the path-polygon is a funnelglass.
6. Compare the results of Step 3 and Step 5. If Step 3 produced a value smaller than twice the length of the shortest path found in Step 5, then return the path-pair corresponding to the best hourglass or bowtie. Otherwise, return the path-pair corresponding to the best funnelglass found in Step 5.

5 Discussion and Analysis

Step 1 of the algorithm builds the visibility graph using the algorithm of Ghosh and Mount [GM]. The structure obtained in this step, which is called the "enhanced visibility graph" by Ghosh and Mount, provides a representation of the nodes visible to a given node sorted angularly about that node. This step takes $O(E + n \log n)$ time and uses $O(E)$ space.

The enhanced visibility graph provides us with the ability to perform Step 2 of the algorithm efficiently; we generate all funnels with base $\overline{s_1 s_2}$ or $\overline{t_1 t_2}$ in $O(E)$ time and $O(E)$ space.

We can represent each funnel with a fixed amount of storage space because a funnel is uniquely determined by the first edge on its lower (or upper) chain when its base is fixed [GM]. This fact also provides a bound of $O(E)$ on the number of funnels with a given base.

Funnels can be generated in output-sensitive time because they are linked to one another in a structure called a funnel tree [GM]. Funnel trees, which are contained implicitly within the enhanced visibility graph, allow us to compute the lengths of each funnel's chains in constant time from the lengths of its parent's chains. A traversal of the "lower" funnel tree produces the lengths of all lower chains and a traversal of the "upper" funnel tree produces the lengths of all upper chains.

In addition to providing us with the length of each funnel in amortized constant time, a funnel tree traversal produces a linear ordering of the funnels called a funnel sequence [GM]. We use the funnel sequence obtained by a clockwise preorder traversal of the lower funnel tree in computing the length of a bowtie.

Step 3 of the algorithm generates hourglasses and bowties between $\overline{s_1 s_2}$ and $\overline{t_1 t_2}$. An hourglass can be partitioned into two funnels by an edge of the visibility graph that is tangent to both defining paths of the hourglass. We choose one of the two such edges to represent the hourglass. This edge uniquely determines the two funnels and, therefore, determines the hourglass and allows us to compute its length from the lengths of the funnels in constant time. So there are $O(E)$ hourglasses and it takes $O(E)$ time to find the best one.

Bowties are processed in a similar fashion. Each bowtie is contained in a unique hourglass and is determined by the same edge as the hourglass that contains it. There are $O(E)$ bowties, too, and the length of each can be determined from the funnel sequence in constant time. Thus we can find the best bowtie in $O(E)$ time.

Finally, the best funnelglass is found by searching the graph constructed in Step 4 of the algorithm. The construction of this graph takes $O(E)$ time since $O(E)$ vertices and edges are added to the visibility graph. The shortest path in this graph can be found in $O(E + n \log n)$ time ([Di],[FT]) and corresponds to the best funnelglass.

In summary, we have the following theorem:

Theorem 3 *An optimal solution to the MIN-SUM problem can be found in $O(E + n \log n)$ time and $O(E)$ space.*

6 The MIN-MAX Problem

In the MIN-MAX problem we want to minimize the *larger* of the two path lengths, rather than the sum of the two path lengths. We use the same basic approach to solve this problem as the one we used for the MIN-SUM problem.

We enumerate funnels, hourglasses, and bowties keeping track this time of each of the two path lengths p_1 and p_2. We now choose the best bowtie or hourglass according to the min-max criterion, but the main difference is that the augmented visibility graph now has two different lengths associated with each edge. We are not able to solve as efficiently the resulting graph search problem since it is a bicriteria shortest path problem. In general, the bicriteria shortest path problem is NP-hard, but our problem has a special structure that allows an efficient solution; namely, the only edges of the graph with two *different* edge lengths are those edges incident to a funnel node.

We find the best funnelglass by enumerating all funnel pairs (with one funnel on base $\overline{s_1 s_2}$ and the other on base $\overline{t_1 t_2}$) and connecting each pair of apices by a shortest path in the visibility graph. After running an all-pairs shortest path algorithm on the visibility graph (in $O(En+n^2 \log n)$ time), we calculate the length of each funnelglass in constant time. Since there are $O(E)$ funnels on each base, it takes $O(E^2)$ comparisons to find the best funnelglass.

In summary, we have the following theorem:

Theorem 4 *An optimal solution to the MIN-MAX problem can be found in $O(E^2 + n^2 \log n)$ time and $O(E)$ space.*

7 The MIN-TIME problem

The MIN-TIME problem is to find a coordinated motion parametrized by time that minimizes the time it takes until both agents reach their destinations, assuming that they both have the same maximum speed. It seems unlikely that our approach can be used to find the optimal solution exactly since min-time paths need not be locally optimal with regard to distance. Therefore, min-time paths need not lie on the visibility graph and will not, in general, form hourglasses, funnelglasses, and bowties. But a simple parameterization of the min-sum path-pair gives a provably good approximation to the optimal solution. We formally state this result below:

Theorem 5 *A simple parameterization of the MIN-SUM path-pair gives an approximate solution to the MIN-TIME problem that takes no more than twice the time of an optimal motion.*

8 Final Remarks

We have shown that optimal solutions to the MIN-SUM and MIN-MAX problems can be computed efficiently. Our algorithm for these problems is, to the best of our knowledge, the first polynomial-time algorithm that finds an optimal motion for a robot system with more than two degrees of freedom.

A more complete account of this work appears in [MW], where we prove the lemmas and theorems stated without proof here and provide historical and algorithmic details not mentioned here.

References

[AAGHI] T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai, "Visibility of Disjoint Polygons", *Algorithmica*, Vol. 1, (1986), pp. 49-63.

[CR] J. Canny and J. Reif, "New Lower Bound Techniques for Robot Motion Planning Problems", *Proc. 28th FOCS*, pp. 49-60, Oct. 1987.

[Di] Dijkstra "A Note On Two Problems in Connection With Graphs", *Numerische Mathematik*, 1 (1959), pp. 269-271.

[FT] M. Fredman and R. Tarjan, "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms", *Proc. 25th Annual IEEE Symposium on Foundations of Computer Science*, pp. 338-346, 1984.

[GM] S.K. Ghosh and D.M. Mount, "An Output Sensitive Algorithm for Computing Visibility Graphs", Technical Report CS-TR-1874, Department of Computer Science, University of Maryland, July 1987. (Also appears in FOCS, 1987.)

[IRWY] C. Icking, G. Rote, E. Welzl, and C. Yap, "Shortest Paths for Line Segments", Technical Report, Fachbereich Mathematik, Freie Universitaet Berlin, September 1989.

[KM] S. Kapoor and S.N. Maheshwari, "Efficient Algorithms for Euclidean Shortest Path and Visibility Problems with Polygonal Obstacles", *Proc. Fourth Annual ACM Symposium on Computational Geometry*, Urbana-Champaign, IL, June 6-8, 1988, pp. 172-182.

[Le] D.T. Lee, "Proximity and Reachability in the Plane", Ph.D. Thesis, Technical Report ACT-12, Coordinated Science Laboratory, University of Illinois, Nov. 1978.

[LP] D.T. Lee and F.P. Preparata, "Euclidean Shortest Paths in the Presence of Rectilinear Boundaries", *Networks*, 14 (1984), pp. 393-410.

[LW] T. Lozano-Pérez and M.A. Wesley, "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles", *Communications of the ACM*, Vol. 22, No. 10 (1979), pp. 560-570.

[Mi] J.S.B. Mitchell, "A New Algorithm for Shortest Paths Among Obstacles in the Plane", Technical Report No. 832, School of Operations Research and Industrial Engineering, Cornell University, October, 1988.

[MW] J.S.B. Mitchell and E.L. Wynters, "Optimal Motion of Covisible Points Among Obstacles in the Plane", Technical Report No. 896, School of Operations Research and Industrial Engineering, Cornell University, March, 1990.

[PS] C.H. Papadimitriou and E.B. Silverberg, "Optimal Piecewise Linear Motion of an Object Among Obstacles", Technical Report, Department of Operations Research, Stanford University, 1986.

[SS] M. Sharir and A. Schorr, "On Shortest Paths in Polyhedral Spaces", *SIAM Journal on Computing* Vol. 15, No. 1, pp. 193-215, February 1986.

[We] E. Welzl, "Constructing the Visibility Graph for n Line Segments in $O(n^2)$ Time", *Information Processing Letters*, Vol. 20 (1985), pp. 167-171.