

# Lazy Robots Constrained by at Most Two Polygons\*

Jeff Abrahamson and Ali Shokoufandeh

Department of Computer Science

Drexel University

Philadelphia, PA 19104

{jeffa, ashokouf}@cs.drexel.com

**Abstract**—We present a polynomial-time algorithm for a special case of the Euclidean traveling salesman problem in which a robot must visit all the vertices of two non-intersecting polygons without crossing any polygon edge. If both polygons are convex, one enclosing the other, our algorithm can find the optimal tour of the channel between them in time  $O(m^3 + m^2n)$  and  $O(nm + m^2)$  space, where the exterior polygon has  $n$  vertices and the interior  $m$  vertices. In the more general case of non-convex polygons (not necessarily nested), the algorithm finds the exact optimum tour in  $O(n^2m + m^3)$  time and  $O(n^2 + m^2)$  space. At the end we give several examples in the context of robot navigation.

**Index Terms**—TSP, Robot Navigation, Computational Geometry.

## I. INTRODUCTION

The Euclidean TSP (ETSP) is the problem of finding a tour of minimum length through a given set of points in  $d$ -dimensional Euclidean space. It is NP hard and not approximable to any polynomial factor [9]. In this paper, we will address a variant of the ETSP in which the points are the vertices of polygons  $P$  and  $Q$ . We assume the edges of  $P$  and  $Q$  are barriers that the tour may not cross. We give a polynomial time algorithm for finding the shortest tour  $T$  through the vertices of  $P$  and  $Q$  while completely avoiding crossing  $P$  and  $Q$ .

Deineko et al. [6] considered a related variant of the ETSP with a convex polygon  $P$  and a set of points on a line segment  $Q$  inside  $P$ . They referred to this problem as the *convex-hull-and-line* ETSP, and gave an  $O(m^2 + mn)$  time and  $O(m + n)$  space algorithm, where  $n$  and  $m$  are the number of  $P$  and  $Q$  vertices, respectively. Cutler [4] gave an  $O(n^3)$  time and  $O(n^2)$  space dynamic programming algorithm for solving the *3-line* ETSP where all points lie on three distinct parallel lines in the plane. Rote [8] extended this result to  $m$ -line ETSP by giving a polynomial dynamic programming algorithm for a fixed number of lines  $m$ . Deineko et al. [5] addressed the problem of recognizing instances of the ETSP for which there is a permutation of points such that the underlying distance matrix fulfills so called Demidenko, Kalmanson and Supnick properties. It is known that if a distance matrix fulfills one of these properties, the ETSP is solvable in polynomial time. We

previously considered a simpler version of this problem [3] in which  $P$  and  $Q$  are both convex. We gave a polynomial time solution to that problem. In an extension to that work [1], [2], we characterized the non-convex case.

In this paper we present a cubic time and square space algorithm for computing the exact solution to this special case of ETSP

## II. DEFINITIONS

In the following we assume a distance function that obeys the triangle inequality and is additive: for all co-linear points  $a$ ,  $b$ , and  $c$  with  $b$  between  $a$  and  $c$ ,  $d(a, c) = d(a, b) + d(b, c)$ . Let  $P$  and  $Q$  denote polygons in the plane. Let  $p_1, \dots, p_n$  denote the vertices of  $P$  and  $q_1, \dots, q_m$  be the vertices of  $Q$ , with  $m, n \geq 3$ . Throughout this paper we will assume that vertices are numbered in the clockwise direction and that subscripts are interpreted modulo  $n$  for  $P$  and  $m$  for  $Q$ . We will also denote two consecutive  $P$  or  $Q$  vertices as  $v$  and  $v^+$ . We consider the exterior of  $P$  and the interior of  $Q$  to be forbidden regions that no valid tour may cross.

An *edge*  $uv$  that connects a pair of points  $u$  and  $v$  is a straight line segment between  $u$  and  $v$  of length  $\|uv\|$ . If the edge between any points  $u$  and  $v$  avoids the forbidden regions then the two points are said to be *visible* to each other. A *path* is a sequence  $u_1, u_2, \dots, u_k$  of vertices and the interconnecting edges  $u_1u_2, u_2u_3, \dots, u_{k-1}u_k$ . We will often write the path  $\pi = u_1 \rightsquigarrow u_k = u_1u_2 \oplus u_2u_3 \oplus \dots \oplus u_{k-1}u_k$ , where  $\oplus$  denotes edge concatenation.

Consider two paths  $\pi_1$  and  $\pi_2$  that share a common vertex  $v$ . We say that  $v$  is a *vertex intersection* if the two paths cross each other at  $v$ . If they touch at  $v$  but do not cross, we call  $v$  a *touch point*. If some edges  $e_1 \in \pi_1$  and  $e_2 \in \pi_2$  intersect, we say that  $\pi_1$  and  $\pi_2$  have an *edge intersection*. Two paths intersect if they have either a vertex or an edge intersection.

If  $\pi = u_1 \rightsquigarrow u_k$  is a path and  $u_1 = u_k$ , then the path is called a *tour*. A *simple* tour  $T$  has no duplicate vertices except the necessary first and last. A tour  $T$  is *weakly-simple* if it has no intersections except possibly for backtracking ( $\dots \oplus vv' \oplus v'v \oplus \dots$ ). Thus, both simple and weakly-simple tours have well-defined interiors.

A tour through all vertices of  $P$  and  $Q$  involves three types of edges: *exterior edges* connecting consecutive  $P$  vertices,

\*Funded in part by a grant from the Office of Naval Research (ONR-N000140410363)

interior edges connecting consecutive  $Q$  vertices, and *cross-over edges* connecting  $P$  vertices with  $Q$  vertices or non-consecutive  $P$  or  $Q$  vertices. (We will see later that non-consecutive  $P$  or  $Q$  vertices can not be adjacent in a shortest tour.) We define  $\| q_i \rightsquigarrow q_j \|$  to be the length of the polygonal path  $q_i \rightsquigarrow q_j = q_i q_{i+1} \oplus \dots \oplus q_{j-1} q_j$ . Note that the path  $q_i \rightsquigarrow q_j$  is clockwise, and that  $(q_i \rightsquigarrow q_j) \oplus (q_j \rightsquigarrow q_i) = Q$ . We similarly define  $p_i \rightsquigarrow p_j$  and its length  $\| p_i \rightsquigarrow p_j \|$ .

Consider a polygon edge  $p_k p_k^+$ . We define a clockwise *detour*  $d_{i,j}^k$  of  $p_k p_k^+$ , for any pair of (not necessarily distinct)  $Q$ -vertices  $q_i$  and  $q_j$  to be the path  $d_{i,j}^k = \mathcal{P}(p_k, q_i) \oplus (q_i \rightsquigarrow q_j) \oplus \mathcal{P}(q_j, p_k^+)$ , where  $\mathcal{P}(u, v)$  denotes a shortest obstacle-avoiding path from  $u$  to  $v$ . Note that the path  $\mathcal{P}(u, v)$  may traverse points of  $Q$  and even result, when added to  $q_i \rightsquigarrow q_j$ , in retracing vertices. Two detours are *disjoint* if the sets of their  $Q$ -vertices are disjoint. The *incremental cost*  $c_{i,j}^k$  of the detour  $d_{i,j}^k$  is

$$c_{i,j}^k = \| \mathcal{P}(p_k, q_i) \| + \| q_i \rightsquigarrow q_j \| + \| \mathcal{P}(q_j, p_k^+) \| - \| p_k p_k^+ \|.$$

Let  $d_{i,j}$  denote a cheapest clockwise detour through  $q_i \rightsquigarrow q_j$  taken over all polygon edges  $p_k p_k^+$ . Set

$$d_{i,j} = \arg \min \{ d_{i,j}^1, d_{i,j}^2, \dots, d_{i,j}^n \}$$

and let

$$c_{i,j} = \min \{ c_{i,j}^1, c_{i,j}^2, \dots, c_{i,j}^n \}.$$

### III. TWO NESTED CONVEX POLYGONS

In this section we provide a structural characterization of a shortest tour  $T$  through the vertices of  $P$  and  $Q$  in the case that  $Q$  is a polygonal obstacle completely enclosed by  $P$ , both polygons convex. The structural characterization will facilitate a transformation of the original problem to  $O(m)$  shortest paths problems in an appropriately defined digraph  $G$ .

In section IV, we remove the convexity and nesting constraints.

#### A. Structural Properties

*Lemma 1:* An optimal tour  $T$  has intersections only on  $Q$  vertices.

(In the interests of brevity, we sometimes give sketches of proofs rather than full proofs where the reasoning is clear. Where only sketches appear, full proofs may be found in [3] and [2].)

**Sketch of Proof:** We consider four cases according to the number of vertices on  $P$ . The result follows from the triangle inequality and careful bookkeeping.  $\square$

*Lemma 2:* There exists a shortest tour  $T$  that visits each  $P$ -vertex exactly once.

**Sketch of Proof:** Suppose not and construct a contradiction by looking at some vertex  $p \in P$  with two visits and using the triangle inequality.  $\square$

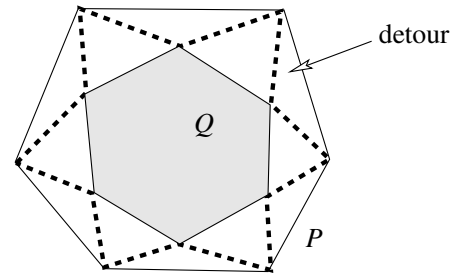


Fig. 1. A case that requires  $\min(m, n)$  detours. Each vertex of  $Q$  is visited by one detour from  $P$ .

*Corollary 3:* There exists a shortest tour  $T$  on which  $P$ -vertices preserve their cyclic order.

In particular, Corollary 3 says that we may orient a tour  $T$  on  $P$ . Henceforth we will consider  $T$  to be clockwise on  $P$  unless otherwise indicated.

Consider a pair of consecutive vertices  $p_k$  and  $p_k^+$  that are not consecutive on a shortest tour  $T$  satisfying the conditions of the above lemmas. From Lemma 1 and the convexity of  $Q$ , then, we may conclude that  $T$  leaves  $p_k$  through a cross-over edge  $p_k q_i$ , visits some  $Q$ -vertices, and returns to  $P$  via a cross-over edge  $q_j p_k^+$ ; i.e.,  $\mathcal{P}(p_k q_i) \oplus q_i \rightsquigarrow q_j \oplus \mathcal{P}(q_j p_k^+)$ , where the path from  $q_i$  to  $q_j$  may be clockwise, counterclockwise, or even retrace some part of its path on  $Q$ . Note that  $q_i$  and  $q_j$  need not be distinct.

Detours in which the orientation on  $Q$  differs from the orientation on  $P$  are special:

*Lemma 4:* If a shortest path is clockwise on  $P$  but counterclockwise on  $Q$ , then  $T$  has precisely one detour from  $P$  to  $Q$ .

*Proof:* By the Jordan curve theorem and Lemma 3, the detour on  $Q$  either covers all of  $Q$  or else it leaves vertices on  $Q$  which can not be reached by  $T$  without crossing the given detour. Since  $T$  visits all vertices of  $P$  and  $Q$ , the single detour covers all of  $Q$ .  $\blacksquare$

It's easy to show, but we do not need the results here, that a shortest tour can be obtained by introducing  $t$  mutually disjoint detours into  $P$ , where  $1 \leq t \leq \min(m, n)$ . The simple example in Figure 1 shows that these bounds are tight. A shortest tour is not always simple, but it must be weakly-simple.

#### B. Algorithm

By Lemma 4, we must consider two cases: the shortest tour with same orientation on  $P$  and  $Q$  and the shortest tour with opposite orientations. Suppose first that the tours have the same orientation.

Let  $G = (V, E)$  be a directed weighted graph with  $V = \{x_1, \dots, x_m, y_1, \dots, y_m\}$  conceptually equal to the vertices of  $Q$  repeated twice, with  $q_i$  represented at  $x_i$  and again at  $y_i$ . The edges  $e(u, v)$  have weight  $w(u, v)$ , where  $w(x_i, y_j) = c_{i,j}$  and  $w(y_i, x_j) = 0$  if  $j = i$  or  $j = i + 1$  and  $\infty$  otherwise,

where subscripts are interpreted modulo  $m$  and  $c_{i,j}$  is the cost of the cheapest detour  $d_{i,j}$ , as defined in Section II. Then one of the  $m$  shortest paths from  $x_i$  to  $y_{i-1}$  corresponds to a shortest obstacle avoiding tour of  $P$  and  $Q$ :

*Theorem 5:* For each  $1 \leq h \leq m$  let  $\pi_h^*$  be the shortest path from  $x_h$  to  $y_{h-1}$ . Let  $\pi^*$  be the  $\pi_h^*$  with shortest length. Then  $\pi^*$  corresponds to a shortest same-orientation obstacle avoiding tour through all  $P$  and  $Q$  vertices.

*Proof:* By Lemmas 1 and 2, the path  $T$  corresponding to  $\pi^*$  is a weakly-simple tour. If a tour of shorter length on the vertices of  $P$  and  $Q$  existed, it would have a representation as a shortest path between some  $x_{h'}$  and  $y_{h'-1}$  in  $G$ . By construction, therefore,  $T$  is the shortest same-orientation tour as claimed. ■

Suppose, on the other hand, that the tour on  $Q$  has opposite orientation than on  $P$ . By Lemma 4, a shortest tour  $T$  will omit precisely one edge of  $P$  and one edge of  $Q$ . We need, therefore, only consider the shortest detour for each candidate edge of  $Q$ .

The shortest tour, then, is the shorter of the shortest same-orientation tour and the shortest opposite-orientation tour.

### C. Complexity

The algorithm in Section III-B leads us down two paths: finding the shortest same-orientation tour and finding the shortest opposite orientation tour. In the former case, we first find the least cost detours for each pair of  $Q$  vertices, then we find  $m$  shortest paths in a graph on  $2m$  vertices. In the latter case we find and compare  $m$  shortest detours.

*Theorem 6:* Finding the shortest tour of  $P$  and  $Q$  vertices requires  $O(m^3 + m^2n)$  time and  $O(nm + m^2)$  space.

*Proof:* Computing the visibility graph of the vertices of  $P$  and  $Q$  takes time and space  $O(mn)$ , since we must consider and store  $mn$  pairs while the supporting tangents can be computed in amortized constant time due to the convexity of  $P$  and  $Q$ .

Since we must consider  $\Theta(nm^2)$  detours, the total cost of computing detours is  $\Theta(nm^2)$ . We use  $O(m^2)$  space to store the values, since we minimize  $c_{i,j}^k$  over  $k$  and so only store the  $c_{i,j}$  and  $d_{i,j}$ .

Finding the shortest same-orientation detours requires  $O(m^3)$  time and  $O(m^2)$  space to find the  $m$  shortest paths (using, for example, the Floyd-Warshall algorithm).

The total complexity is thus  $O(m^3 + m^2n)$  time and  $O(nm + m^2)$  space.

Finding the shortest opposite-orientation tour, from the above, requires  $O(nm + m^2n)$  time and  $O(mn + m^2)$  space, since we must still compute the visibility graph and the  $d_{i,j}$ 's.

To find the shortest tour, we must find each of the above, and the result follows. ■

## IV. TWO ARBITRARY NON-INTERSECTING POLYGONS

We now consider the case where the convexity constraint is removed.  $P$  and  $Q$ , then, are arbitrary, simple, non-

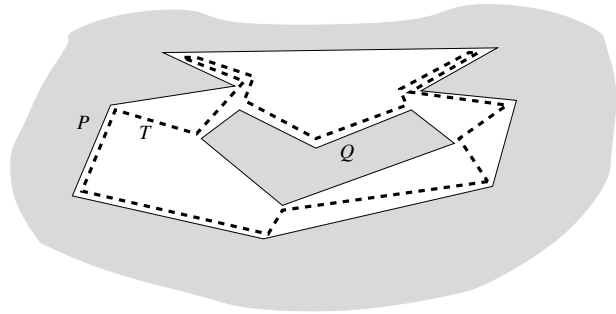


Fig. 2. A shortest tour might visit some  $P$  and  $Q$  vertices more than once.

intersecting polygons with  $Q$  contained within  $P$ . At the end of this section we will additionally remove the requirement that  $Q$  lie within  $P$ .

### A. Structural Properties

In the current case that  $P$  and  $Q$  may be non-convex, we define  $d_{i,j}^k$  and  $c_{i,j}^k$  as earlier, but note that  $\mathcal{P}(u,v)$  may result in backtracking on both  $P$  and  $Q$ .

Let  $A$  be a polygon and denote by  $\text{Convex}(A)$  its convex hull. Let  $a_i$  and  $a_j$  be two consecutive points on  $\text{Convex}(A)$  that are not consecutive on  $A$  itself. We call the points  $a_i, a_{i+1}, \dots, a_j$  together with the segments connecting them a bay of  $A$ . We say that the segment  $a_i a_j$  of the convex hull crosses the bay. Cf. Figure 3.

In this section we provide a structural characterization of a shortest tour  $T$  through the vertices of  $P$  and  $Q$ . It will again facilitate a transformation of the original problem to  $O(m)$  shortest paths problems.

*Lemma 7:* An optimal tour  $T$  has intersections only on  $P$  and  $Q$  vertices.

**Sketch of Proof:** The proof follows similar reasoning to the arguments given earlier in Section III with the addition of intermediate points when points are not visible. □

It would appear that for an optimal solution to the problem of finding a tour through the vertices of polygons  $P$  and  $Q$  the tour must visit each vertex exactly once. In III we showed that in the case of  $P$  and  $Q$  convex, some tours that visit some vertices of  $Q$  more than once may be shorter than tours that visit each  $Q$  vertex precisely once. In the case of arbitrary  $P$  and  $Q$ , moreover, even  $P$  vertices may be visited more than once in a shortest tour, as Figure 2 shows.

When  $P$  is convex, optimal solutions can be characterized as consisting of detours to  $Q$  vertices between appropriately chosen pairs of consecutive  $P$  vertices. As Figure 2 shows, this is no longer the case when  $P$  and  $Q$  are not convex. A similar result is still true, however:

*Corollary 8:* There exists a shortest tour  $T$  on which  $P$  vertices preserve their cyclic order after all sub-tours of the form  $\dots pp'p \dots$  are removed.

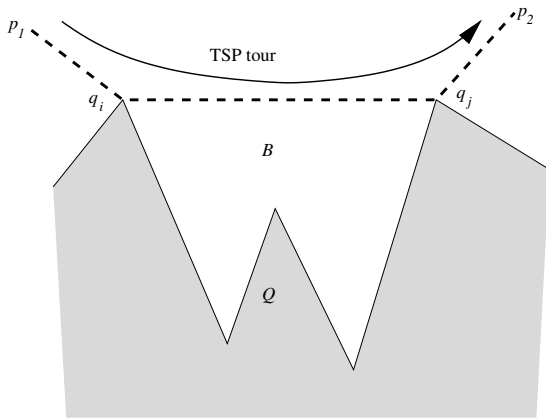


Fig. 3. In the special case that  $P$  is convex, the case that the optimal tour of  $P \cup \text{Convex}(Q)$  crosses a bay.

That is, if we remove backtracking on  $P$ , the order of vertices on  $P$  is strictly cyclic, although we miss the backtracked vertices we skipped.

*Proof:* Assume the contrary. Then the tour must intersect itself in the space between  $P$  and  $Q$ , which is forbidden by Lemma 7. ■

Detours in which the orientation on  $Q$  differs from the orientation on  $P$  are still special by a variant on the same proof as in [3]:

**Lemma 9:** If a shortest path, ignoring backtracking, is clockwise on  $P$  but counterclockwise on  $Q$ , then  $T$  has precisely one detour from  $P$  to  $Q$ .

*Proof:* By the Jordan curve theorem and Lemma 8, the detour on  $Q$  either covers all of  $Q$  or else it leaves vertices on  $Q$  which can not be reached by  $T$  without crossing the given detour. Since  $T$  visits all vertices of  $P$  and  $Q$ , the single detour covers all of  $Q$ . ■

When  $P$  is convex, we can partially compensate for the concavity of  $Q$  by computing the shortest tour on  $P \cup \text{Convex}(Q)$  and repairing the bay crossings.

**Lemma 10:** Let  $P$  be a convex polygon and  $Q$  a simple polygon contained within  $P$ . Let  $T$  be a shortest tour on  $P \cup \text{Convex}(Q)$ . Suppose that some segment of  $T$  crosses a bay  $B \subset Q$ . Then there exists a shortest tour  $T'$  on  $P \cup Q$  that follows the points of  $B$ .

**Sketch of Proof:** If not, the tour through the bay would cross itself or leave the bay and return, both of which would be longer than staying in the bay. □

Unfortunately, the related structural lemma we would need to extend our results directly—that is, by fixing each bay—does not hold, even in the case that  $P$  is convex. Suppose  $T$  is a shortest tour on  $P \cup \text{Convex}(Q)$  and that some segment of  $T$  returns to  $P$  rather than cross a bay  $B \subset Q$  (Figure 4). Fixing  $B$  is not in general any easier than solving the general channel ETSP, as Figure 5 shows. Because repairing bay crossings doesn't work in general, we do not consider the special case

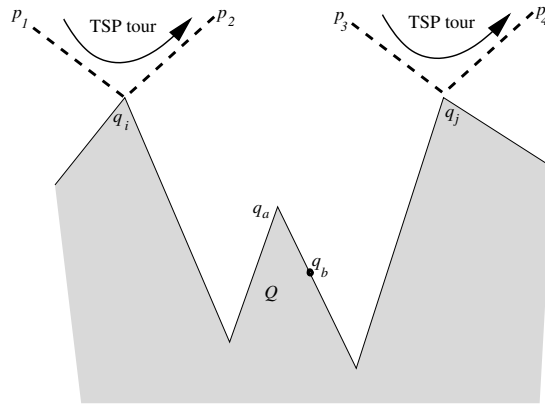


Fig. 4. In the special case that  $P$  is convex, the case that the optimal tour of  $P \cup \text{Convex}(Q)$  does not cross some bay.

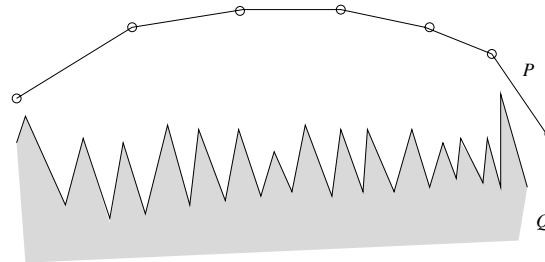


Fig. 5. A bay with  $O(m)$  peaks and  $O(n)$  vertices in  $P$  above..

of convex  $P$  further.

## B. Algorithm

Let  $P$  and  $Q$  be arbitrary simple polygons with  $Q$  entirely in the interior of  $P$ . In [3] we computed minimum detours  $d_{i,j}$  from some  $p_k$  to  $q_i$  to  $q_j$  returning to  $p_k^+$ . We were able to ignore the cost of the implicit minimization in the  $\mathcal{P}(p, q)$  terms, since the convexity of  $P$  and  $Q$  guaranteed that the furthest visible point in each direction moved monotonically. Without convexity, we must compute the set of visible points anew for each vertex  $p \in P$ . Indeed, some points of  $P$  may be visible from no point of  $Q$  and vice versa.

In practice we can be clever by computing the convex hull of  $Q$  and noting that no point inside (meaning not protruding from) a bay of  $Q$  both of whose endpoints are not visible to some  $p$  is itself visible to  $p$ . Nonetheless, in the worst case (cf. Figure 5),  $O(m)$  vertices of  $Q$  may be visible to  $O(n)$  points of  $P$ .

By Lemma 9 we must again consider the two cases of same orientation and opposite orientation. Suppose first that the tour has the same orientation on  $P$  and  $Q$ .

Let  $G = (V, E)$  be a directed weighted graph as in Section III. Then one of the  $m$  shortest paths from  $x_i$  to  $y_{i-1}$  again corresponds to a shortest obstacle avoiding tour of  $P$  and  $Q$ :

Theorem 5 holds in the context of non-convex polygons as well:

*Theorem 11:* For each  $1 \leq h \leq m$  let  $\pi_h^*$  be the shortest path from  $x_h$  to  $y_{h-1}$ . Let  $\pi^*$  be the  $\pi_h^*$  with shortest length. Then  $\pi^*$  corresponds to a shortest same-orientation obstacle avoiding tour through all  $P$  and  $Q$  vertices.

By Lemma 9, a shortest opposite-orientation tour  $T$  will again omit precisely one edge of  $P$  and one edge of  $Q$ . We need, therefore, only consider the shortest detour for each candidate edge of  $Q$ .

The shortest tour, then, is again the shorter of the shortest same-orientation tour and the shortest opposite-orientation tour.

### C. Complexity

The algorithm in Section IV-B leads us down two paths: finding the shortest same-orientation tour and finding the shortest opposite orientation tour. In the former case, we first find the least cost detours for each pair of  $Q$  vertices, then we find  $m$  shortest paths in a graph on  $2m$  vertices. In the latter case we find and compare  $m$  shortest detours.

*Theorem 12:* Finding the shortest tour of  $P$  and  $Q$  vertices requires  $O(n^2m + nm^2 + m^3)$  time and  $O(n^2 + m^2)$  space.

*Proof:* The cost of computing the set of visible points for each  $p \in P$  is  $O(m(n+m))$  [7], and so  $O(nm(n+m))$  for all  $p$ . Computing the points of  $P$  visible to vertices of  $Q$  is trivial, on the other hand, once we have computed  $Q$ 's visibility from  $P$ , since the visibility graph is undirected. Computing visibility thus costs  $O(nm(n+m))$ . The space requirements are  $O((n+m)^2)$  to store the edges of the graph.

Since we must consider  $\Theta(nm^2)$  detours, the total cost of computing detours is  $\Theta(nm^2)$ . We use  $O(m^2)$  space to store the values, since we minimize  $c_{i,j}^k$  over  $k$  and so only store the  $c_{i,j}$  and  $d_{i,j}$ .

Finding the shortest same-orientation detours requires  $O(m^3)$  time and  $O(m^2)$  space to find the  $m$  shortest paths (using, for example, the Floyd-Warshall algorithm).

The total complexity is thus  $O(nm^2 + m^3 + nm(n+m)) = O(n^2m + nm^2 + m^3)$  time and  $O((n+m)^2 + m^2) = O(n^2 + m^2)$  space for the same-orientation shortest-tour.

To find the shortest opposite-orientation tour we compute the visibility graph and then compare  $m$  detours to  $Q$  for each of  $n$  adjacent vertices of  $P$ , using  $O(nm)$  time and constant space (since we only need store the shortest). The total cost is thus  $O(nm(n+m) + m^3) = O(n^2m + nm^2 + m^3)$  time and  $O((n+m)^2) = O(n^2 + m^2)$  space. ■

We promised at the beginning of this section to remove the nesting constraint of  $Q$  within  $P$ . In fact, we did not use the convexity in the above discussion. Indeed, the only reason for the slightly sly unneeded assumption of nesting is that the words ideas of same and opposite orientation have reversed meanings in the nested and un-nested case. The algorithm therefore works unmodified in the case of two non-nested polygons.

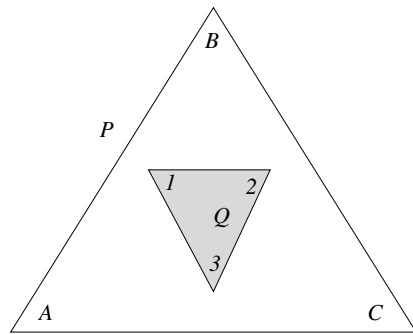


Fig. 6. A triangular obstacle inside another triangle.

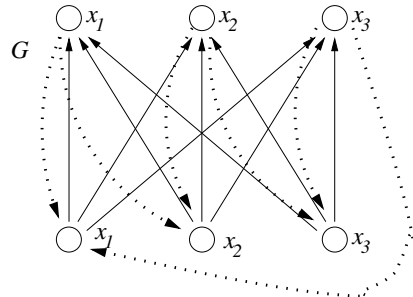


Fig. 7. The shortest path graph from a triangular obstacle inside another triangle. The vertices  $x_1, x_2$ , and  $x_3$  represent the three vertices of  $Q$  twice. The solid lines indicate the  $c_{i,j}$ , the dotted lines indicate the zero weight return edges.

## V. EXPERIMENTS

Consider Figure 6, a simple arrangement of two polygons: a rotated triangle embedded in another triangle. This example is sufficiently simple to visualize the graph  $G$  in an understandable way, Figure 7. The shortest tour is clearly that in Figure 8.

We can also use our algorithm to compute the optimal tour of any closed channel without loops, as Figure 9 shows. Here we have merely taken the two ends of the channel (blue) and connected them with a narrow and nearly featureless passage, creating two closed curves: the polygons  $P$  and  $Q$ . Since the

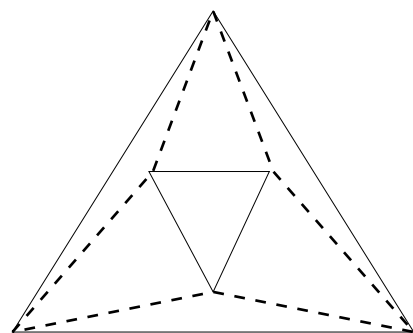


Fig. 8. The shortest path visiting the vertices of two triangles.

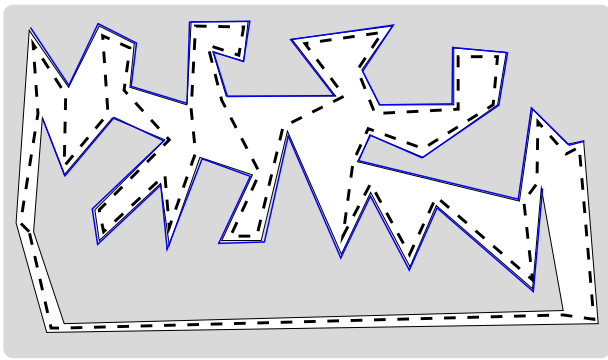


Fig. 9. The shortest tour along an irregular corridor. Note that we form the channel by wrapping the ends around so that they form two polygons.

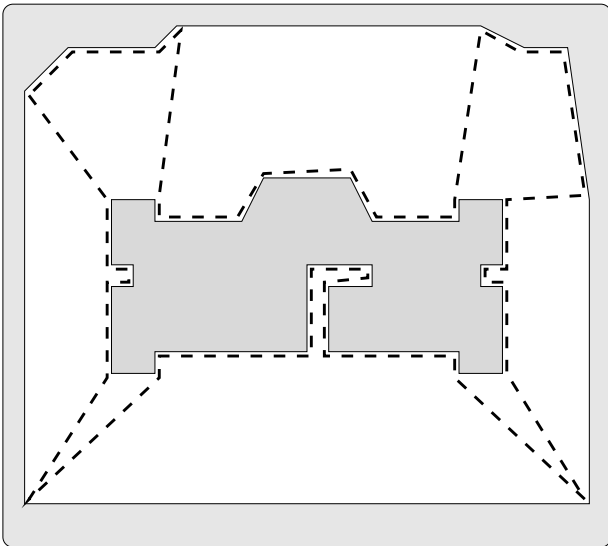


Fig. 10. The shortest tour of watchpoints around a building (with service garage) and the perimeter of its property.

tour of the connecting passage is trivial and doesn't affect the one-way tour of the channel, the result is an optimal tour of the channel that visits every vertex along its length.

As a final example, consider patrolling the exterior of a building and the perimeter of its property, Figure 10. The robot is to visit every corner along the property line as well as every corner of the building. Since we have made no assumption of the angle at a vertex, note that points on edges may be designated as vertices without affecting the algorithm at hand. Our use of the word "corner," then, can be taken quite liberally.

## VI. FUTURE WORK AND CONCLUSIONS

It would be nice to extend our results on the Euclidean TSP to the case of more than two polygons. The techniques used in this paper, unfortunately, are not readily extensible to more than  $k = 2$  polygons. In particular, one might hope to place a third polygon around the existing two, but the

technique of computing detours is not obviously adaptable to the third polygon.

If we could find an algorithm to treat cases  $k > 2$ , we can still make a few statements about those algorithms. First, the case  $k > 3$  only makes sense if no more than one polygon encloses others, for the obstacle nature of the polygons that characterizes this version of the problem would make the problem insoluble if some polygon separated the set of polygons into two non-empty sets.

Second, the computational complexity must clearly increase without bound with  $k$ , for in the limiting case that each polygon approaches a point relative to the inter-polygon distances, this becomes the classic ETSP problem in its full generality.

Nonetheless, it may be possible, using techniques similar to those in this paper, to solve  $k = 3$  in the special case that polygons  $Q_1$  and  $Q_2$  are completely enclosed within  $P$  and the convex hull of  $Q_1 \cup Q_2$  does not intersect  $P$ . If one could do this, the result could probably be extended further by the same trick we use at the end of this paper to transform the enclosing polygon into another non-enclosing polygon. As noted above, however, the reduction to ordinary ETSP would limit the usefulness of continuing in this manner.

## ACKNOWLEDGMENT

The authors thank Pawel Winter of the University of Copenhagen for suggesting the initial idea for this problem. We also thank M. Fatih Demirci of Drexel University for his helpful suggestions.

## REFERENCES

- [1] Jeff Abrahamson. Between a rock and a hard place: Euclidean TSP in the presence of polygonal obstacles. Master's thesis, Drexel University, Department of Computer Science, jun-aug 2005.
- [2] Jeff Abrahamson, Ali Shokoufandeh, and Pawel Winter. ETSP on a polygon in the presence of a polygonal obstacle. Technical report, Drexel University, 2004.
- [3] Jeff Abrahamson, Ali Shokoufandeh, and Pawel Winter. Euclidean TSP between two nested convex obstacles. *Information Processing Letters*, To appear.
- [4] M. Cutler. Efficient special case algorithms for the  $n$ -line planar traveling salesman problem. *Networks*, 10:183–195, 1980.
- [5] V. G. Deineko, R. Rudolf, J. A. A. van der Veen, and G. J. Woeginger. Three easy special cases of the euclidean traveling salesman problem. Technical Report 17, Technische Univ., Graz, Austria, 1995.
- [6] V. G. Deineko, R. van Dal, and G. Rote. The convex-hull-and-line traveling salesman problem: a solvable case. *Information Processing Letters*, 51:141–148, 1994.
- [7] M. de-Berg et al. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, 1998.
- [8] G. Rote. The  $n$ -line traveling salesman problem. *Networks*, 22:92–108, 1992.
- [9] V. Vazirani. *Approximation Algorithms*. Springer, Berlin, 2001.