

Euclidean TSP on Two Polygons

Jeff Abrahamson

*Dept. of Computer Science
Drexel University
Philadelphia, PA
jeffa@cs.drexel.edu
USA*

Ali Shokoufandeh*

*Dept. of Computer Science
Drexel University
Philadelphia, PA
ashokouf@cs.drexel.edu
USA*

November 15, 2005

Abstract

We give an $O(n^2m + nm^2 + m^2 \log m)$ time and $O(n^2 + m^2)$ space algorithm for finding the shortest traveling salesman tour through the vertices of two simple polygonal obstacles in the Euclidean plane, where n and m are the number of vertices of the two polygons.

Keywords: Computational Geometry; Traveling Salesman Problem; Convex Analysis; Network Flow

1 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. In this paper, we address a variant of the ETSP in which the points are the vertices of two polygons P and Q . We assume the tour may not cross P or Q . We give a polynomial time algorithm for finding the shortest tour T through the vertices of P and Q while completely avoiding crossing between the interior and exterior of either polygon.

Deineko et al. [3] 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. Cutler [2] 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. Deineko et al. [4] 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. Their algorithm runs in $O(m^2 + mn)$ time and $O(m + n)$ space, where n and m are the number of P and Q vertices, respectively. Rote [9, 8] extended this result to m -line ETSP by giving a polynomial dynamic programming algorithm for a fixed number of lines m . We previously considered a related version of this problem [1] in which P and Q are both convex

*This research was supported in part by the Office of Naval Research grant ONR-N000140410363.

and nested. We gave an $O(m^2 \log m + m^2 n)$ time and $O(nm + m^2)$ space exact solution to that problem. In this paper we generalize these results to the case of two arbitrary simple polygonal obstacles, achieving running time of $O(n^2 m + nm^2 + m^2 \log m)$ time and $O(n^2 + m^2)$ space.

2 Basic Definitions

In the following we assume a distance function that obeys the triangle inequality and is additive: $d(a, c) = d(a, b) + d(b, c)$ for all co-linear points a , b , and c with b between a and c .

Let P and Q denote simple polygonal obstacles in the plane with well defined non-empty interiors. We assume throughout this paper that the two polygons do not intersect each other or themselves. Let p_0, \dots, p_{n-1} denote the vertices of P and q_0, \dots, q_{m-1} be the vertices of Q , with $m, n \geq 3$. Throughout this paper we will assume that vertices are numbered in the clockwise direction if P and Q are nested, else clockwise on P and counterclockwise on Q if P and Q are not nested. We further assume 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^+ .

An *edge* uv , also called a *polygon edge*, that connects a pair of points u and v is a straight line segment between u and v of length $d(u, v)$. If the interior of the line segment between points u and v avoids crossing any edge or vertex of P and Q , then we say that the two points *visible* to each other. A *path* is a sequence u_1, u_2, \dots, u_k of vertices along with their interconnecting edges $u_1 u_2, u_2 u_3, \dots, u_{k-1} u_k$. We will often write the path $\pi = u_1 \rightsquigarrow u_k = u_1 u_2 \oplus u_2 u_3 \oplus \dots \oplus u_{k-1} u_k$, where \oplus denotes concatenation. Note that a path may contain vertices of both P and Q . A path that does not cross between the interior and exterior of either polygon we call an *obstacle avoiding path*. Let $\pi_u = u_1 \rightsquigarrow u_k = u_1 u_2 \oplus u_2 u_3 \oplus \dots \oplus u_{k-1} u_k$ and $\pi_v = v_1 \rightsquigarrow v_l = v_1 v_2 \oplus v_2 v_3 \oplus \dots \oplus v_{l-1} v_l$ be two paths. We call $\phi_u(t)$ a *parameterization* of π_u if $\phi_u : [0, 1] \rightarrow \pi_u$ is a surjective mapping of the closed interval $[0, 1] \subset \mathbb{R}$ onto the path π_u such that $\phi_u(0) = u_1$ and $\phi_u(1) = u_k$. Suppose $\phi_u(t)$ and $\phi_v(t)$ are parameterizations of π_u and π_v respectively. We say that π_u and π_v *intersect* if there exist $s, t \in [0, 1]$ such that $\phi_u(s) = \phi_v(t)$. Otherwise we say that they do not *non-intersect*.

If $\pi = u_1 \rightsquigarrow u_k$ is a path and $u_1 = u_k$, then the path is called a *tour*. We will see later that tours proceed in straight line segments from one polygonal vertex to another. With malice of foresight, then, we call such a line segment a *tour edge*. Usually when we speak of edges it will be clear from context whether we mean tour edges or polygonal edges and so we will not usually specify.

We would like to define the seemingly intuitive concept of a tour backtracking on itself, see Figure 1. For the purposes of this paragraph, then, consider a total ordering on the polygonal edges of P , $p_0 p_1 < p_1 p_2 < \dots < p_{n-2} p_{n-1} < p_{n-1} p_0$, and similarly on Q , $q_0 q_1 < q_1 q_2 < \dots < q_{n-2} q_{n-1} < q_{n-1} q_0$. We call a polygonal edge *increasing* with respect to a given tour if the tour traverses the edge only in order of increasing vertex index. A tour edge is said to be increasing on a polygonal edge if it traverses the polygonal edge in order of increasing vertex index. We define decreasing polygonal and tour edges similarly. We call a tour *increasing* on P if the first polygonal edge of P under the total ordering above that is traversed by the tour in only one direction is increasing. If the first edge of P traversed in only one direction is decreasing, we say the tour is *decreasing* on P . We will see later that such a characterization is reasonable, but here we use it solely for the purpose of the following definition: a tour edge of P is a *backtracking edge* if the tour is increasing on P and the tour edge is decreasing or if the tour is decreasing on P and the tour edge is increasing. We make the parallel definitions and observations for Q . Note that we have defined increasing (and decreasing) separately for polygonal edges, for tour edges, and for tours.

A *simple* tour T has no duplicate vertices except the necessary first and last. A tour T is *weakly-simple* if it

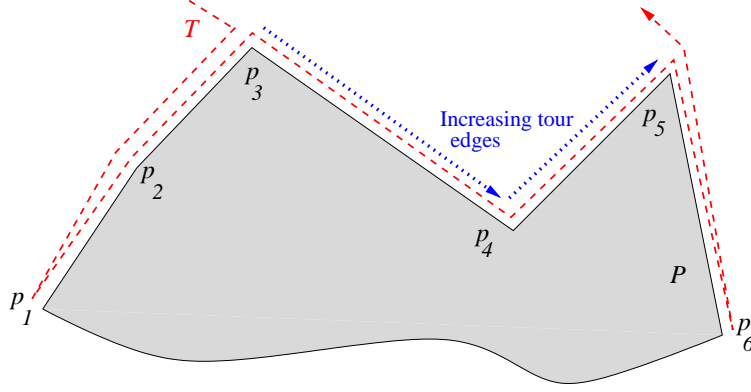


Figure 1: Part of a tour T showing two increasing tour edges on P . The tour T is therefore increasing on P . The tour backtracks from p_3 to p_1 and again from p_6 to p_5 .

has no duplicate vertices except possibly for backtracking. Thus, both simple and weakly-simple tours have well-defined interiors. We say that a tour T *intersects itself* if no parameterization $\phi_T(t)$ of T is invertible.

A tour through all vertices of P and Q involves two types of edges: *polygonal edges* connecting consecutive P vertices or 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 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 *detour*, cf. Figure 2, $d_{i,j}^k$ of $p_k p_k^+$, for any pair of not necessarily distinct Q vertices q_i and q_j to be the path $d_{i,j}^k = \mathcal{S}(p_k, q_i) \oplus (q_i \rightsquigarrow q_j) \oplus \mathcal{S}(q_j, p_k^+)$, where $\mathcal{S}(u, v)$ denotes a shortest obstacle-avoiding path from u to v of length $\|\mathcal{S}(u, v)\|$.

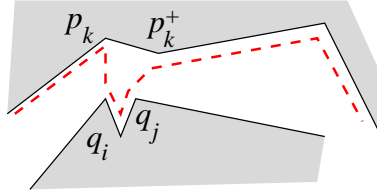


Figure 2: The detour $d_{i,j}^k$ from p_k to p_k^+ through $q_i \rightsquigarrow q_j$.

Note that the path $\mathcal{S}(u, v)$ may traverse points of P and Q and even result, when added to adjacent segments of the tour sections before u or after v , in retracing vertices, cf. Figure 3. Two detours are *disjoint* if the sets of their P and Q vertices are disjoint. The *incremental cost* $c_{i,j}^k$ of the detour $d_{i,j}^k$ is

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

Let $d_{i,j}$ denote the cheapest detour through $q_i \rightsquigarrow q_j$ taken over all polygonal edges $p_k p_{k+}$. In case of more than one minimum cost $d_{i,j}^k$, we choose the $d_{i,j}^k$ with least k . Note that $c_{i,j}$ is the cost of $d_{i,j}$.

Let A be a polygon and denote by $\text{Convex}(A)$ the smallest convex polygon containing A . Let a_i and a_j be two consecutive points on $\text{Convex}(A)$ with $j \neq i + 1$. We call the path $a_i \rightsquigarrow a_j$ a *pocket* of A . We say that the segment $a_i a_j$ of the convex hull *crosses the pocket*. Cf. Figure 4.

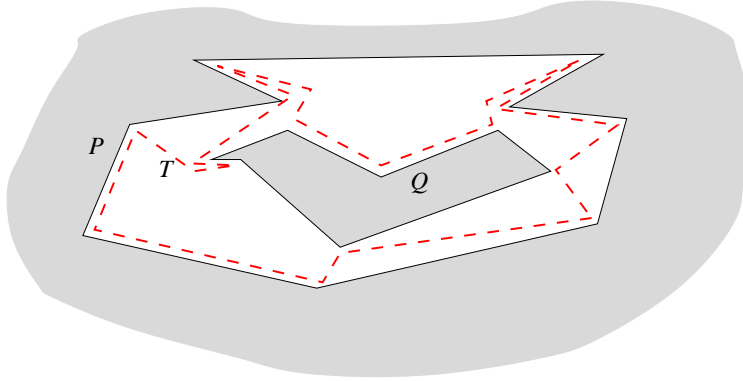


Figure 3: A shortest tour might visit some P and Q vertices more than once.

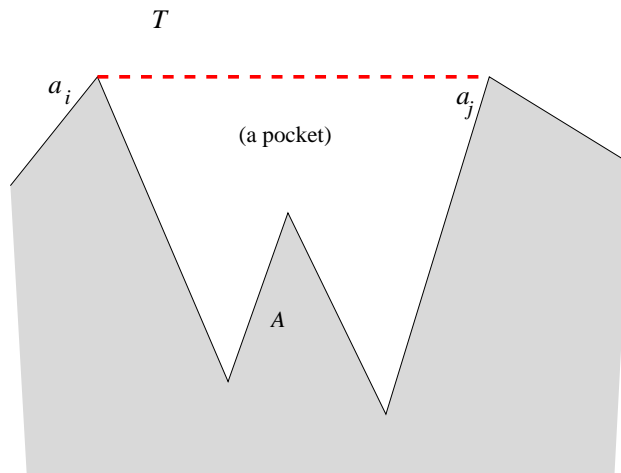


Figure 4: The tour T crossing a pocket of A .

Since tours on vertices of nested polygons are somewhat easier to visualize than on non-nested, the figures we use to illustrate our arguments will often show nested polygons. Note that our arguments do not assume nesting.

3 Structural Properties

In this section we provide a structural characterization of a shortest tour T through the vertices of P and Q . It will facilitate a transformation of the original problem to m shortest paths problems in an appropriately defined digraph.

We often implicitly use the following seemingly intuitive result about a convex chain in a triangle, cf. Figure 1:

Proposition 1. *Let abc be a triangle and C a convex chain in its interior between a and c of length $\|C\|$. Then $d(a, b) + d(b, c) > \|C\|$.*

Proof. The proof is by repeated application of the triangle inequality. Suppose x is some point in the interior of abc such that C is in the interior of the triangle axc and such that either ax or xc is tangent to a segment of the chain C . Clearly showing that $d(a, x) + d(x, c) < d(a, b) + d(b, c)$ is sufficient to prove the proposition.

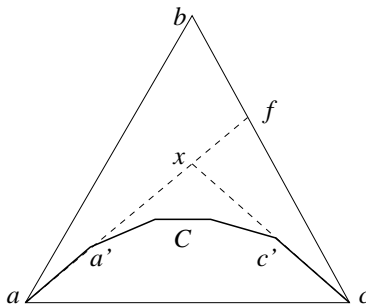


Figure 5: The construction from Proposition 1. C is a convex chain in triangle abc .

Extend the line segment ax until it intersects edge bc and call the point of intersection f (Figure 5). Then by the triangle inequality we have that $d(a, b) + d(b, c) > d(a, f) + d(f, c) > d(a, x) + d(x, c)$. If C has more than two segments, it is always possible to choose x such that at least one segment of C is on ax or xc . We may therefore iterate the procedure above by transforming abc and C into $a'xc'$ and C' where $C' \subset C$ connects a' to c' and has fewer segments than C . Repeating this procedure we remove at least one segment of the chain at each step. The procedure therefore eventually terminates in a comparison of the triangle abc with a two segment (triangular) chain. \square

Lemma 2. *An optimal tour T has intersections only on P and Q vertices.*

Proof. Assume to the contrary that an optimal tour T contains an intersection in the region between P and Q . We need only consider the following five cases. Note that the vertices are necessarily distinct, since otherwise we would merely be describing backtracking. Each of the five cases will lead to a contradiction, proving the Lemma.

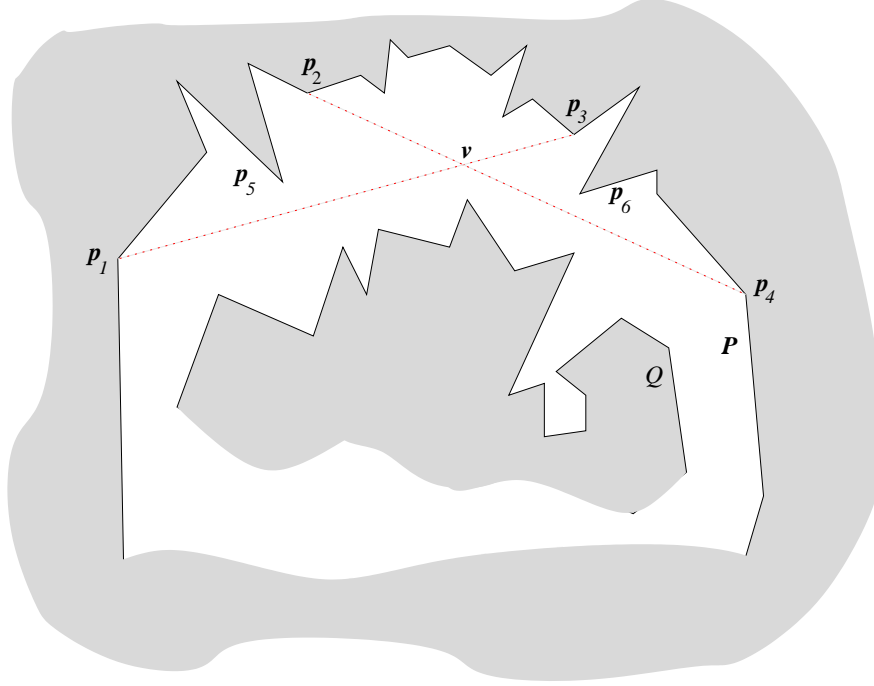


Figure 6: In Lemma 2, replacing p_1p_3 and p_2p_4 with p_1vp_2 and p_3vp_4 results in a tour of the same length. Relaxing to $p_1p_5p_2$ and $p_3p_6p_4$ results in a shorter tour.

1. **All four vertices are on P .** Call the four points p_1 , p_2 , p_3 , and p_4 and denote their point of intersection v . Suppose the shortest tour contains segments p_1p_3 and p_2p_4 , as shown in Figure 6. Then replacing p_1p_3 and p_2p_4 with either p_1vp_2 and p_3vp_4 or with p_1vp_4 and p_2vp_3 results in a tour of the same length, one of which must still be connected. Relaxing the tour away from v (cf. Figure 6) results, by Proposition 1, in a tour no longer than the original.
2. **Three vertices are on P and one on Q .** Let p_1, p_2 and p_3 denote the three vertices on P and q the vertex on Q such that v is the point of intersection of p_1p_3 and p_2q , as in Figure 7. We may substitute either p_1vq and p_2vp_3 or p_1vp_2 and qvp_3 without increasing the length of the tour. Relaxing the tour away from v to the convex hull of the points in the triangle again results in a tour of no greater length.
3. **Two vertices are on P and two on Q .** Let p_1, p_2, q_1 , and q_2 denote the four vertices and suppose p_1q_2 and p_2q_1 intersect at a point v . We may substitute either p_1vq_1 and p_2vq_2 without increasing the length of the tour. Observe that substituting p_1vp_2 and q_1vq_2 is not possible without intersecting P or Q . There exists, therefore, paths $p_1 \rightsquigarrow q_1$ and $p_2 \rightsquigarrow q_2$ (Cf. Figure 8) that are shorter by Proposition 1.
4. **One vertex is on P and three are on Q .** By symmetry, this case is equivalent to Case 2.
5. **All four vertices are on Q .** By symmetry, this case is equivalent to Case 1.

□

The relaxation techniques in the above proof lead immediately to

Corollary 3. *Let T be a shortest tour on the vertices of P and Q without crossing any edge or vertex of P or Q . Then any vertex that T visits more than once is due to backtracking.*

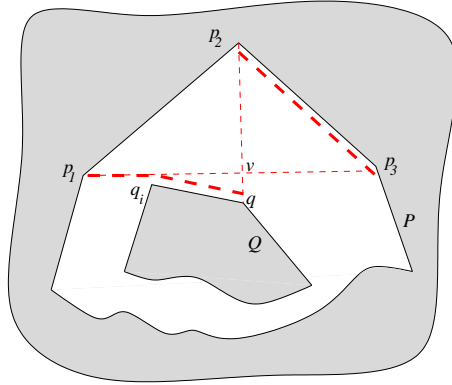


Figure 7: In lemma 2, case 2, three vertices on P , using a detour across Q . The bold dashed line shows a shorter tour section than the original non-bold dashed tour section.

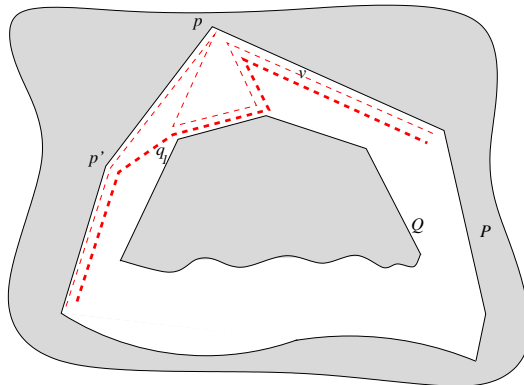


Figure 8: In Lemma 2, case 3, the bold dashed line represents a shorter tour section than the original non-bold dashed tour section.

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 fact, the tour may backtrack as it avoids obstacles, and so a shortest tour may visit vertices more than once, as Figure 3 illustrates.

As we shall see, it would be very convenient to characterize shortest tours as sets of detours to Q from a cyclic traversal of P . Figure 3 illustrates that this is not possible. Nevertheless, we can so characterize shortest tours if we are willing to omit those vertices visited during backtracking. Indeed, backtracking is perhaps best viewed as following graph edges that simply happen to trace over other vertices, as does the long edge of a triangle on three collinear points.

Corollary 4. *There exists a shortest tour T on which P vertices preserve their cyclic order after all backtracking edges are removed.*

That is, if we ignore backtracking edges of the tour on P , the order of the remaining vertices on P is strictly cyclic. Note that by symmetry the same is true on Q .

Proof. Assume the contrary. Then the tour must have a subtour that is clockwise on P and a subtour on P that is counter-clockwise. Since the path can not cross P (that is, either the interior or the exterior of P is forbidden), then even without consideration of Q it follows that the tour must cross itself, which is forbidden by Lemma 2 and Corollary 3. \square

Since a shortest tour is cyclic on the vertices of P as well as cyclic on the vertices of Q , we can talk about a tour having the same or different orientations on P and Q . We will say that a tour is a *same orientation tour* if its restriction to P and Q respectively is cyclic following increasing vertex number, as for example in the two instances shown in Figure 9. Otherwise we will say that the tour is an *opposite orientation tour*.

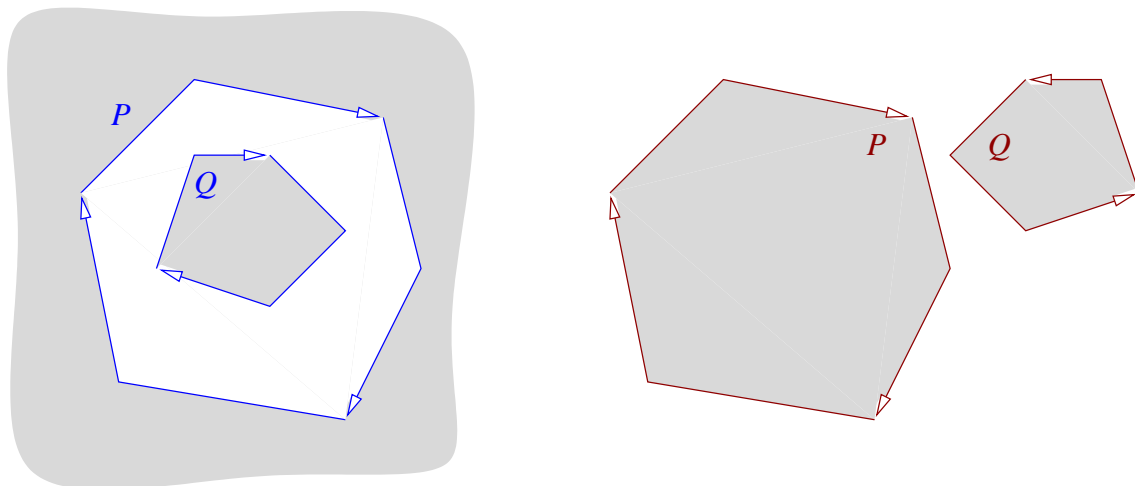


Figure 9: Two same orientation tours: nested (left) and non-nested (right). Recall that our convention for vertex numbering is different for nested and non-nested polygons.

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

Lemma 5. *If a shortest path, ignoring backtracking, proceeds by increasing (modulo n) vertex index on P but decreasing (modulo m) vertex index on Q , then T has precisely one detour from P to Q .*

Proof. By the Jordan curve theorem [7] and Lemma 4, 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 . \square

One is tempted to use the results of [1] on the convex hulls of P and Q and then to “fix” the pockets. (We shall see shortly that this approach eventually fails.) When P is convex, reads the temptation, we can partially compensate for the concavity of Q by computing the shortest tour on $P \cup \text{Convex}(Q)$ and repairing the pocket crossings. So far this is possible:

Lemma 6. *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 pocket $B \subset Q$. Then there exists a shortest tour T' on $P \cup Q$ that follows the points of B .*

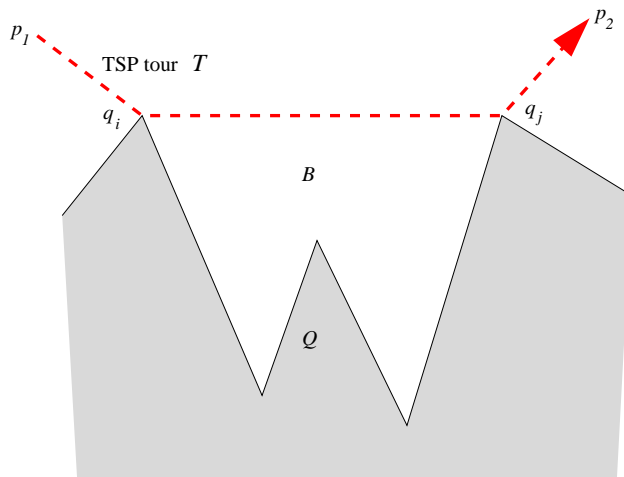


Figure 10: In the special case that P is convex, the case that the optimal tour of $P \cup \text{Convex}(Q)$ crosses a pocket.

Proof. Suppose without loss of generality that the pocket B and the tour T are labeled as in Figure 10. The structural lemmas of [1] tell us that p_1 and p_2 must be adjacent on P , justifying our vertex labeling in Figure 10. Suppose that some tour T on $P \cup Q$ is shorter than any tour that follows $p_1 q_i q_{i+1} \dots q_j p_2$. Suppose that T enters B at some point q_k . Then $q_i \rightsquigarrow q_{k-1}$ must return to P , which is a contradiction by Corollary 3. \square

It is not clear, however, how to derive the related structural lemma we would need to extend our results directly by fixing each pocket. For example, 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 pocket $B \subset Q$, as in Figure 11. In general the structure of P and Q between q_i and q_j may be arbitrarily complex. It is not obvious how to exploit the pocket structure and the shortest tour on $P \cup \text{Convex}(Q)$ to simplify this problem. We will see in Section 4 that we can proceed in the absence of a repair strategy. Because of this, we do not consider further the special case of convex P .

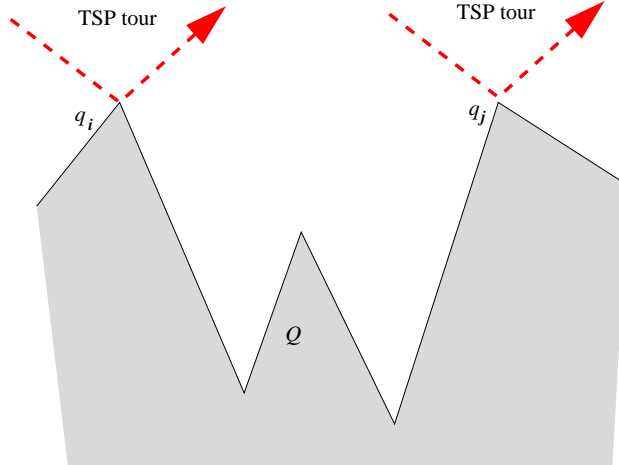


Figure 11: If P is convex and optimal tour of $P \cup \text{Convex}(Q)$ does not cross some pocket, we are no better off for the simplifying assumptions, for the structure of Q inside the pocket and P between q_i and q_j may be arbitrarily complex.

4 Algorithm

The structural characterizations of the previous section now allow us to present and analyze an algorithm to compute a shortest tour of the vertices of two polygons. By Lemma 5 we must consider two cases: the shortest tour with same orientation on P and Q and also the shortest tour with opposite orientation. Suppose first that the tour has the same orientation on P and Q .

Let $G = (X, Y, E)$ be a complete directed weighted bipartite graph with bipartition $X = \{x_0, \dots, x_{m-1}\}$ and $Y = \{y_0, \dots, y_{m-1}\}$ conceptually equal to the vertices of Q repeated twice, with q_i represented at x_i and again at y_i , $i \in \{0, 1, \dots, m-1\}$. The directed edge (u, v) has weight $w(u, v)$, where $w(x_i, y_j) = c_{i,j}$ and $w(y_i, x_j) = 0$ if $j = i + 1$ and ∞ otherwise. See Figure 12 for a simple example involving a triangle and a square. As noted earlier, subscripts are interpreted modulo m and $c_{i,j}$ is the cost of a cheapest detour $d_{i,j}$, as defined in Section 2. Observe that this construction presents a one-to-one correspondence between same orientation tours on the vertices of P and Q obeying the structural theorems of Section 3 and paths in G from x_h to y_{h-1} for some h . In particular, by Corollary 4 each shortest tour corresponds to a cyclic traversal of P with detours to Q . Since all vertices of Q must be represented in a valid tour, a shortest path in G , viewed as a list of detours from P to Q , completely describes a same-orientation tour. That is, a path in G from x_h to y_{h-1} describes a set of detours from a cyclic traversal of P and Q that visits every vertex of Q . Similarly, given a tour, the list of detours from P to Q identifies a shortest path through the graph G . One of the m shortest paths from x_i to y_{i-1} , then, corresponds to a shortest obstacle avoiding tour of P and Q :

Theorem 7. *For each $0 \leq h < m$ let Π_h be the least weight path from x_h to y_{h-1} . Let Π be the Π_h with least weight. Then Π corresponds to a shortest same-orientation obstacle avoiding tour through all P and Q vertices.*

Proof. Let Π be as above, and call h^* a value of h for which $\Pi = \Pi_{h^*}$. Denote by T_{h^*} a corresponding tour through the vertices of P and Q . Clearly T_{h^*} is a valid tour, since it is a cyclic traversal of P with detours to Q that is obstacle avoiding and that visits every vertex of P and Q . The detour $d_{i,j}$ is represented in the graph G by edge (x_i, y_j) . The continuation along P until the next detour is represented by one of the

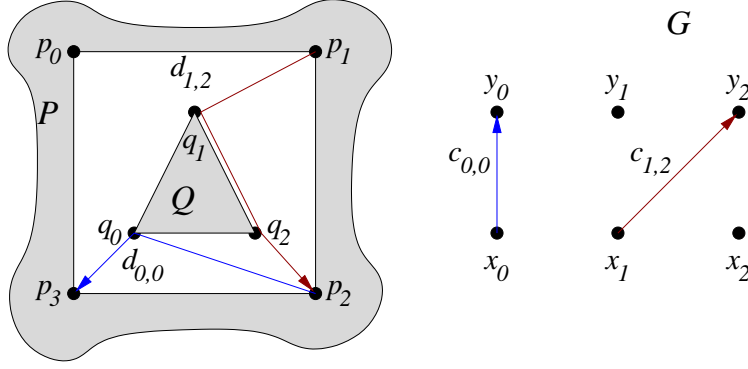


Figure 12: Two polygons P (a square) and Q (a triangle) and the vertices of the associated graph G . The edge (x_0, y_0) corresponds to the detour $d_{0,0}$ on the vertices of P and Q , the least cost detour from some vertex p of P to vertex q_0 and back to vertex p^+ of P . Similarly, edge (x_1, y_2) corresponds to the detour $d_{1,2}$ on the vertices of P and Q . Conversely, detour $d_{0,0} = p_2q_0 \oplus q_0p_3$ with cost $c_{0,0}$ and $d_{1,2} = p_1q_1 \oplus q_1q_2 \oplus q_2p_2$ with cost $c_{1,2}$ are represented in the graph G by the edges (x_0, y_0) and (x_1, y_2) respectively.

zero weight edges from y_j to x_{j+1} . Suppose T_h were a tour on the vertices of P and Q with length less than that of T_{h^*} . Then T_h would have a representation as a path in G from x_h to y_{h-1} lighter than Π_{h^*} , which is a contradiction. By construction, therefore, T_{h^*} is the shortest same-orientation tour as claimed. It has length $|P| + |\Pi_{h^*}|$. \square

Suppose, on the other hand, that the tour on Q has opposite orientation than on P . By Lemma 5, 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:

Corollary 8. *Let P and Q be simple polygons as above. Then the shortest obstacle-avoiding tour of the vertices of P and Q is found by taking the shorter of the shortest tour obtained from Theorem 7 and the shortest opposite-orientation tour.*

5 Complexity

The algorithm in Section 4 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 9. *Finding the shortest tour of P and Q vertices requires $O(n^2m + nm^2 + m^2 \log m)$ time and $O(n^2 + m^2)$ space.*

Proof. The cost of computing the set of visible points for each $p \in P$ is $O(m(n+m))$ [5], 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 $O(m^2 \lg m)$ time and $O(m^2)$ space to find the m shortest paths [6]. (Note that Karger et al.'s $m^* = O(m)$ in our application.) The total complexity is thus $O(m^2 \lg m + m^2 n)$ time and $O(nm + m^2)$ space.

The total complexity is thus $O(nm^2 + m^2 \lg m + nm(n + m)) = O(n^2 m + nm^2 + m^2 \lg m)$ 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 $O(nm(n + m) + m^2 \lg m) = O(n^2 m + nm^2 + m^2 \lg m)$ time and $O((n + m)^2) = O(n^2 + m^2)$ space. \square

6 Convex Obstacles and Other Shortcuts

If Q is nested inside P and both polygons are convex, then Abrahamson et al. [1] provide a slightly more efficient algorithm. In fact, we will see that nesting is not required to gain these efficiencies. Computing visibility by calipers [10] results in complexity of $O(m^2 \lg m + m^2 n)$ time and $O(nm + m^2)$ space. The following corollary is therefore a generalization of that work with similar complexity.

Corollary 10. *Let P and Q be convex polygons. Finding the shortest tour of P and Q vertices that does not cross Q requires $O(m^2 \lg m + m^2 n)$ 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 using calipers due to the convexity of P and Q [10].

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 $O(m^2 \lg m)$ time and $O(m^2)$ space to find the m shortest paths [6]. The total complexity is thus $O(m^2 \lg m + m^2 n)$ time and $O(nm + m^2)$ space.

Finding the shortest opposite-orientation tour, from the above, requires $O(nm + m^2 n)$ 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 the shortest same-orientation tour and the shortest opposite orientation tour, and the result follows. \square

If Q is nested inside P , P convex and Q flattened to a line segment, Deineko et al. [4] showed an $O(mn)$ time and $O(n)$ space algorithm for computing a least cost tour, where n is the total number of vertices of which m lie on the line Q .

7 Non-planar Tours

A reasonable follow-on question regards extension to higher dimension. The answer, if not the proof, is short.

Theorem 11. *Let P and Q be non-intersecting polytopes in a three dimensional vector space X with additive distance function that obeys the triangle inequality. If $P \neq NP$, computing a shortest TSP tour on the vertices of P and Q while considering the polytopes as obstacles is NP hard.*

Proof. We reduce to the planar case. Let G be a complete graph in the $z = 0$ plane (X with the z dimension projected out and under the same distance function restricted to the plane). Throughout this proof we will assume edges have weight equal to their length in X .

Choose d_{\min} such that

$$0 < d_{\min} \ll \min \left(\min_{e \in E} w(e), \min_{\substack{e_i, e_j \in E \\ e_i \neq e_j}} |w(e_i) - w(e_j)| \right)$$

and choose $d_{\max} > |E| \max_{e \in E} w(e)$. Choose ϵ such that $d_{\min}/|E| \gg \epsilon > 0$.

We will now construct (cf. Figures 13 and 14) a polytope P from G and a second polytope Q whose only purpose is to fit a second polytope to the problem without changing the shortest tour on the vertices of P . We will construct the polytope P such that finding a shortest tour on P and Q will find, as a subproblem, a shortest tour on G . With this reduction we will have proved that the problem is NP hard.

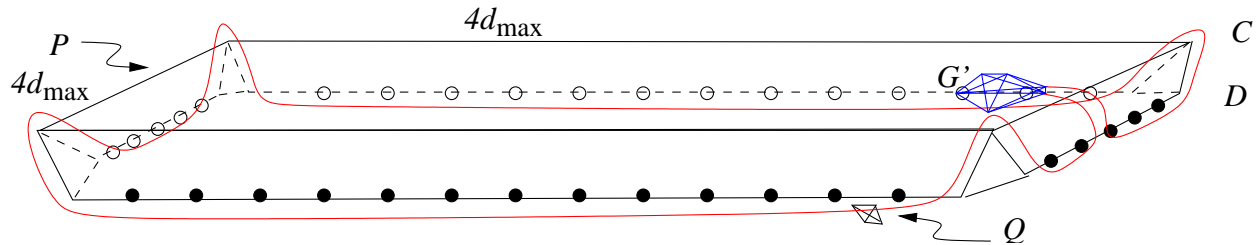


Figure 13: The construction of non-nested 3-D polytopes P and Q . The traveling salesman tour (red) is free to visit any vertex of P and Q in any order as long as the vertices are visible to each other. By construction, the tour visits the vertices of G (blue) separately from the non- G vertices.

We need to consider two essentially equivalent reductions: the case that Q is nested inside P and the non-nested case. Consider first the non-nested case. We begin with the graph G . We will construct a graph P that will define a polytope, which, by abuse of notation, we will also call P .

Start with $P = G$. We will add vertices (and so edges) to form our final polytope.

Choose some vertex $b \in V$ such that b is on the convex hull of V . Replace b by two distinct vertices, b_1 and b_2 , in the $z = 0$ plane such that $|b_1 - b_2| < \epsilon$, $b_1 = b$, and b_2 is on the convex hull of V . We will call this augmented graph $G' = G - \{b\} + \{b_1, b_2\}$. By our definition of ϵ and straightforward application of the triangle inequality, we have the following:

Claim 12. *T' is a shortest tour on the augmented graph G' if and only if an identical tour T with ub_1b_2v changed to ubv on the original graph G is a shortest tour. In particular, a shortest tour on G' visits b_1 and b_2 consecutively.*

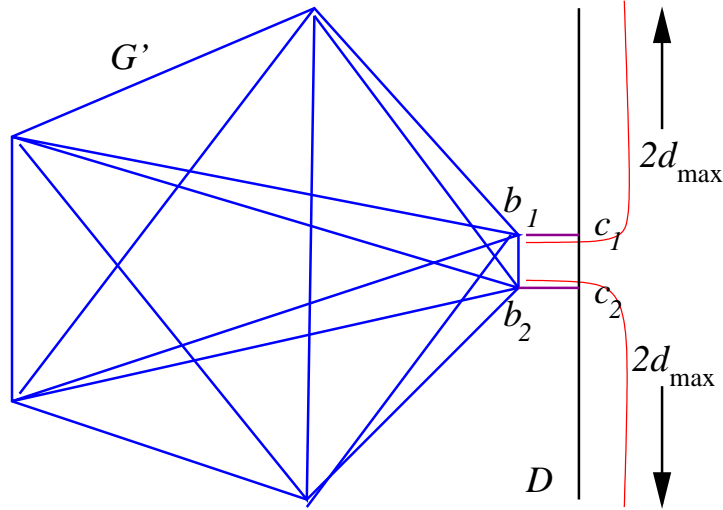


Figure 14: Detail of G' and the square $b_1b_2c_2c_1$ in the construction of non-nested 3-D polytopes P and Q . Here the traveling salesman tour is red, the augmented graph G' is blue, the upper square D is black.

We continue with the construction. In the $z = 0$ plane add two points c_1 and c_2 such that $b_1b_2c_2c_1$ is a square and c_1 and c_2 are outside the convex hull of V , cf. Figure 14. Then add four additional points, the vertices of square C with sides of length $4d_{\max}$ such that c_1c_2 sits in the middle of one of the sides of the square and G lies entirely in the interior of C . Finally, consider the square D with truncated corners in the $z = -\epsilon$ plane with edge length $4d_{\max} - 2\epsilon$ centered under C . Add $4|E|$ points to P distributed uniformly on the interiors of the edges of D . The result, Figure 13, is a square pad with truncated bottom corners and with G embedded in its top face.

For the polytope Q , create a regular tetrahedron with edge lengths $\epsilon/2|E|$, positioned outside P a distance $\epsilon/2|E|$ from one of the sides of D , as indicated in Figure 13.

Claim 13. *A shortest tour T of $P \cup Q$ consists of a sub-tour T_1 of the vertices of G and a sub-tour T_2 of the vertices of $C \cup D \cup Q = (P \cup Q) - G$. Moreover, the transition between the two sub-tours T_1 and T_2 occurs on the line segments b_1c_1 and b_2c_2 .*

Proof. Let T be a shortest tour on $P \cup Q$. By construction T will traverse the points of D in cyclic order with one or more detours to points of C and precisely one detour to points of Q . By construction, furthermore, the single detour to Q must occur only from the nearest vertices of D to Q . Moreover, any detour from $C \cup D$ to G other than at the $b_1c_1c_2b_2$ square would increase the cost of the tour by more than any savings from a change in tour order on $V(G)$, since a shortest tour on G has length less than d_{\max} . \square

With this characterization of shortest tours on $P \cup Q$ and Claim 12, it follows that a shortest tour on $P \cup Q$ computes as a sub-tour a shortest tour on G , which can not be done in polynomial time if $P \neq NP$. Therefore T can not be found in polynomial time unless $P = NP$.

The nested reduction is the same as the non-nested reduction except that we place Q inside P . \square

Note that no dimension higher than 3 can have a polynomial time solution, since the higher dimensions

could be made thin enough that a solution in $d > 3$ dimensions could be used to find a solution in three dimensions.

8 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 technique. As noted above, however, the reduction to ordinary ETSP would limit the usefulness of continuing in this manner.

To the best of our knowledge this is the first polynomial algorithm for this problem. The more general case of k polygons (for fixed $k \geq 2$) remains open. No lower bound for the problem is known beyond the trivial $O(n \log n)$ time needed to compute a cyclic ordering of the vertices.

Acknowledgments

The authors thank Pawel Winter of the University of Copenhagen for suggesting the initial idea for this problem. This work was funded in part by a grant from the Office of Naval Research (ONR-N000140410363).

References

- [1] Jeff Abrahamson, Ali Shokoufandeh, and Pawel Winter. Euclidean TSP between two nested convex obstacles. *Information Processing Letters*, 95:370–375, 2005.
- [2] M. Cutler. Efficient special case algorithms for the n -line planar traveling salesman problem. *Networks*, 10:183–195, 1980.
- [3] 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.

- [4] 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.
- [5] M. de-Berg et al. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, 1998.
- [6] D. R. Karger, D. Koller, and S. J. Phillips. Finding the hidden path: time bounds for all-pairs shortest paths. *SIAM Journal of Computing*, 22:1199–1217, 1993.
- [7] James Munkres. *Topology: A First Course*. Prentice-Hall, 1975.
- [8] G. Rote. *Two solvable cases of the Traveling Salesman Problem*. PhD dissertation, Technische Universität Graz, Institut für Mathematik, 1988.
- [9] G. Rote. The n -line traveling salesman problem. *Networks*, 22:92–108, 1992.
- [10] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proceedings of IEEE MELECON*, Athens, Greece, may 1983.