Touring a Sequence of Polygons in Weighted Regions

Amir Hedayaty*
Department of Computer Engineering
Sharif University of Technology

Salman Parsa†
Department of Mathematical Scince
Sharif University of Technology

Mohammad Ghodsi‡
Department of Computer Engineering
Sharif University of Technology
IPM School of Computer Science

Abstract

Given a subdivision of plane into convex polygon regions, a sequence of polygons to meet, a start point s, and a target point t, we are interested in determining the shortest weighted path on this plane which starts at s, visits each of the polygons in the given order, and ends at t. The length of a path in weighted regions is defined as the sum of the lengths of the sub-paths within each region. We will present an approximation algorithm with maximum δ cost additive. Our algorithm is based on the shortest weighted path algorithm proposed by Mata and Mitchel [2]. The algorithm runs in $O(((n^3LW + RW)\frac{k}{\delta})^3)$ time, where n is the number of vertices of the region boundaries, L is the longest boundary, W is the maximum weight in the region, R is the sum of the perimeters of the regions, and k is the number of polygons. The main idea in the algorithm is to add Steiner points on the region boundaries and polygon edges. In addition, we will also present a solution to the query version of this problem. We will extend our result in unweighted version of the "Touring a Sequence of Polygons" problem [3]. We will give an approximation algorithm to solve the general case of the problem (with non-convex intersecting polygons).

Keywords computational geometry, shortest path, weighted region, polygons

1 Introduction

1.1 Problem Definition

In computational geometry there are several problems (e.g., the safari problem, the watchman route problem, and the zoo-keepers problem) involving touring a sequence of the polygons. Although, these problems have been solved with efficient and optimal algorithms, variants of the problems involving weighted regions are still considerable. To model a weighted subdivision, we can consider that as some polygonal regions with a weight associated to each region. If the number of vertices in the subdivision are n, then the number of polygons will be $O(n)$. By adding edges between reflex vertices, we can divide non-convex regions into convex ones with the same weights. Since the number of vertices is still n, the number of regions will remain $O(n)$. This lets us model the weighted plane as a number convex polygon regions with each region associated a weight α.

In this paper we will examine new vari-

*hedayaty@ce.sharif.edu
†parsa@math.sharif.edu
‡ghodsi@sharif.edu, This author's work has been partly supported by IPM School of CS (contract: CS1385-2-01).

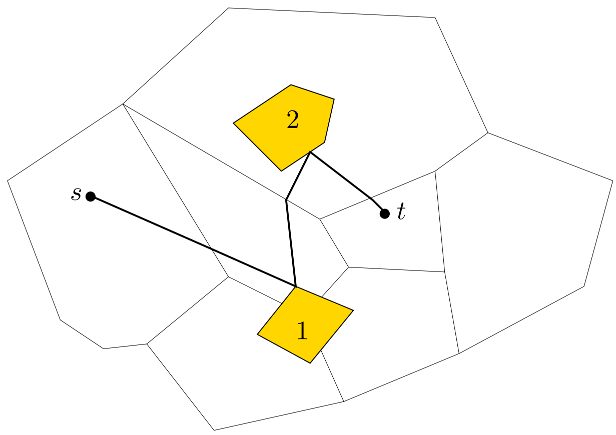

Figure 1: Shortest weighted path meeting two polygons.

ants of two problems “Touring a Sequence of Polygons” (TPP) and “The Weighted Region Problem” (WRP). In TPP, given a sequence of k polygons, a starting point(s), and a target point(t), we are asked to find the shortest path which starts from s and ends in t and meets the given polygons in the given order. In the case when the order of polygons is not given, the problem is proved to be NP-HARD by reducing the metric TSP. The answer for metric TSP (Traveling Salesman Problem) is the answer for the problem if we consider the polygons as just points. Dror, Efrat, Lubiw, and Mitchell have presented the optimal result for the problem in [3] with the running time of $O(kn\log(n/k))$ when the polygons are convex and disjoint.

WRP is a generalization of the shortest path problem in the plane with obstacles, in which we assume that polygon is subdivided into convex polygon regions each of which is associated with a weight α specifying the cost per unit distance of traveling in that region. One can easily observe that the weight of each edge is the minimum of neighbor weights. The goal of WRP is to find the weighted shortest path for two given points. WRP was first studied by Papadimitriou and Mitchell [1] where they presented an $O(n^8\log(n/\epsilon))$ running time $(1+\epsilon)$-algorithm. Later Mata and Mitchel in [2] presented other algorithm with $O(n^3/\epsilon)$ running time. Minimum angle and ratio of maximum weight to minimum weight have also an effect on the running time of the algorithm. In the same paper they have presented another algorithm based on edge subdivision that discretize the problem by adding Steiner points to the edges. The algorithm results to an additive approximation with an error of at most δ with a running time of $O(\frac{LWn^2\log n}{\delta})$. The best result for the problem is given by Aleksandrov, Maheshwari, and Sack in [5]. They have placed $O((1/\epsilon)\log(1/\epsilon))$ points on each of the edges of regions. By a modified version of Dijkestra's algorithm they propose a $O((n/\epsilon)(1/\sqrt{\epsilon}+\log n)\log(1/\epsilon))$ time algorithm.

1.2 Our Approach

In this paper our goal is to find the shortest path from s to t meeting the polygons in the given order, and having the minimum weighted cost as illustrated in Figure 1.

We will also restrict the result for unweighted regions and approximately solve a general case of the TPP which polygons can intersect and be non-convex (which Dror et al in [3] have proved to be NP-HARD) with the cost of being approximation algorithm with high running time.

We will use several parameters in this paper. W is the maximum weight in the plane. L is the maximum edge in the plane. k is the number of polygons to meet. R is the

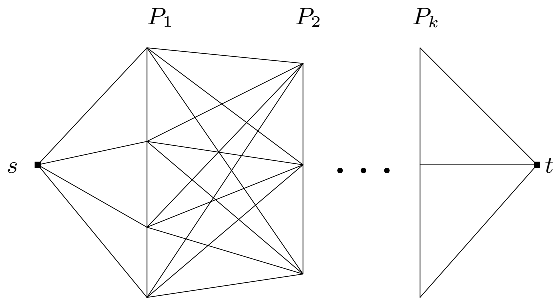

Figure 2: A WSP graph.

sum of the perimeters of the polygons. M is the maximum perimeter of the polygons. We don't care about the number of vertices of polygons. n is the number of region polygon vertices. The algorithm will generate an additive approximation solution with the cost at most δ more than the optimal solution cost.

This paper follows with our algorithm for TPP on weighted regions in Section 2. In Section 3 we will analyze the running time of the algorithm and we will also present error analysis to prove the approximation. Next, we will present the query version of the algorithm in Section 4. Section 5 will include applications of the presented algorithms. Finally, section 6 includes usability of the algorithms, possible future works, and summary of results.

1.3 Summary of Results

1. An approximation algorithm with $O(((n^3LW + RW)\frac{k}{\delta})^3)$ running time for the touring a sequence of polygons in weighted regions.

2. An approximation algorithm with $O(((n^3LW + RW)\frac{k}{\delta})^3)$ preprocess time and $O(M^2W^2k^3/\delta^2)$ query time for touring a sequence of polygons problem in weighted regions, where query can include start and target point, a subset of polygons and order of them.

3. An approximation algorithm with $O(M^2k^3/\delta^2)$ running time for the general case of touring a sequence of polygons.

4. Solving touring a sequence of polygons in weighted regions will result solving "Zoo Keepers", "Watchman Route" and "Safari" problems in weighted regions.

2 TPP in Weighted Regions

The main idea in our algorithm is adding c_e/δ' Steiner points evenly spaced on each of the regions and polygons edges, where c_e is the cost of edge. Later we will show that $\delta' = \delta/3(k+1)n^2$ is enough for region edges and $\delta' = \delta/3(k+1)$ is enough for polygon edges. This will place points with distance at most $\delta/3(k+1)n^2$ on region edges and with distance $\delta/3(k+1)$ on polygon edges. Next we will join the Steiner points inside each of the weighted regions and construct a weighted graph and solve the problem inside the graph. First we will compute all-pairs shortest path. Then, we will construct a weighted graph WSP in which s is connected to all to point in the first polygon with the shortest calculated path between them as weight of the edges. And all point on the first polygon are connected to all points on the second polygon with the shortest path between them as the weight as the weight of the edges. All the points on the last polygon are connected to t with the shortest path between them as weight the weight of the edges. The WSP is illustrated in Figure 2. We will use all-pairs shortest path algorithm to calculate the weights for WSP graph. Then we will use a dynamic algorithm to evaluate the shortest path between s and t in WSP. The idea is to calculate the shortest path from s to all of points in P_1, then calculate the shortest path from s to all of the points on P_{i+1}. Thus for each point p in P_{i+1} we need to find the point p' in P_i where the distance from s to p' plus the distance from p' to p is minimum. We need to check each of the edges of WSP once, so the running time of the algorithm will be $O(|E(WSP)|)$. In next section we will show that there exists a path in WSP with cost at most δ more than the optimal solution cost.

3 Analysis

First we will perform an error analysis and then running time analysis. Suppose that $\pi^*(s,t)$ is the shortest weighted path from s to t with polygon meet constraints. Let p_i be the point at which $\pi^*(s,t)$ meets polygon P_i. We use the notation $d'(u,v)$ for the weight between u and v in WSP. Also $d(u,v)$ means the distance between u and v in π^* if they belong to π^*. We use the term $\mathrm{SP}(u,v)$ for the optimal shortest path between u and v. Since the cost between points on each polygon is at most $\frac{\delta}{3(k+1)}$, there is a Steiner point p'_i on P_i which has distance at most $\frac{\delta}{3(k+1)}$ to p_i.

Lemma 1 $d'(u,v) < \mathrm{SP}(u,v) + \frac{\delta}{3(k+1)}$.

Proof. Lemma 7.1 on [1] proves that the optimal shortest path between any two points in a given weighted region with n vertices is at most $O(n^2)$. Since there are points with distance $\delta/3(k+1)n^2$ at the boundaries of the regions for each point in optimal path, there is a Steiner point with distance at most $\delta/(3k+1)n^2$. Starting from a source point choosing a Steiner point close to the optimal path, when we reach the target point we will find a path trough Steiner points with distance at most $\delta/3(k+1)$ with the optimal path. So if we choose any two points $u, v \in WSP$, there is a path trough Steiner points from u to v with cost at most $SP(u,v) + \frac{\delta}{3(k+1)}$. Thus $d'(u,v) < \mathrm{SP}(u,v) + \frac{\delta}{3(k+1)}$. □

Lemma 2 $d'(s,p'_i) < d(s,p_i) + \frac{2}{3}\frac{\delta}{(k+1)}$

Proof. By triangular inequality we know that $\mathrm{SP}(s,p'_1) < d(s,p_1) + \frac{\delta}{3(k+1)}$. By Lemma 1 we know that the $d'(s,p'_1) < \mathrm{SP}(s,p'_1) + \frac{\delta}{3(k+1)}$. So we will have $\mathrm{SP}(s,p'_1) < d(s,p'_1) + \frac{2}{3}\frac{\delta}{(k+1)}$. □

Lemma 3 *There is a point p'_{i+1} on P_{i+1} where $d'(p'_i,p'_{i+1}) < d(p_i,p_{i+1}) + \frac{\delta}{(k+1)}$*

Proof. There is a point p'_{i+1} on P_{i+1} which has a distance of $\frac{\delta}{3(k+1)}$ from p_{i+1}. By using triangular inequality twice, we will have $\mathrm{SP}(p'_i,p'_{i+1}) < d(p_i,p_{i+1}) + \frac{2}{3}\frac{\delta}{(k+1)}$, and by applying Lemma 1, we will have $d'(p'_i,p'_{i+1}) < d(p_i,p_{i+1}) + \frac{\delta}{(k+1)}$. □

Lemma 4 $d'(p'_k,t) < d(p_k,t) + \frac{2}{3}\frac{\delta}{(k+1)}$

Proof. Similar to Lemma 2, by triangular inequality we know that $\mathrm{SP}(p'_k,t) < d(p_k,t) + \frac{\delta}{3(k+1)}$. By Lemma 1 we know that the $d'(p'_k,t) < \mathrm{SP}(p'_k,t) + \frac{\delta}{3(k+1)}$. So we will have $\mathrm{SP}(p'_k,t) < d(p'_k,t) + \frac{2}{3}\frac{\delta}{(k+1)}$. □

Lemma 5 *There is a path π' in WSP which has length at most $\pi^* + \delta$.*

Proof. We will choose π' the path starting from s, traversing through p'_is, and ending in t. We will have:

$$d'(s,p'_1) < d(s,p_1) + \frac{2}{3}\frac{\delta}{(k+1)}$$

$$d'(p'_i,p'_{i+1}) < d(p_i,p_{i+1}) + \frac{\delta}{(k+1)}$$

$$d'(p'_k,t) < d(p_k,t) + \frac{2}{3}\frac{\delta}{(k+1)}$$

$$d'(s,p'_1) + \sum_{i=1}^{k-1} d'(p'_i,p'_{i+1}) + d'(p'_k,t) <$$

$$d(s,p_1) + \sum_{i=1}^{k-1} d(p_i,p_{i+1}) + d(p_k,t) + \delta$$

So $\pi' < \pi^* + \delta$ which will complete the proof. □

Since π is the shortest path inside WSP, we have $\pi \leq \pi'$ for any π'. Merging the results we will have:

$$\pi < \pi' < \pi^* + \delta$$

.

For the running time analysis we first count the number of vertices. The number Steiner points on the region edges is $O(n^3kLW/\delta)$. The number of Steiner points on the polygons are $O(RWk/\delta)$. So total number of vertices are $V = O((n^3LW + RW)\frac{k}{\delta})$. Running all-pairs shortest path result $O(V^3)$ running time. The dynamic programming part is $O(|E(WSP)|)$. $|E(WSP)|$ can be bounded to $O(M^2W^2k^2/\delta^3)$ in worst case. Usually this is negligible by $O(V^3)$. The overall running time of the algorithm will become $O(V^3 + Mk^3/\delta)$. If we change the way weights of WSP are computed (in fact we need just the edges of WSP not all pairs) as followed, the running time of the algorithm will be $O(VM^2W^2k^3 \log V/\delta^2)$. For computation of each of the edges of WSP, we perform the Dijkestra algorithm once. This algorithm will run faster than the former algorithm for small M.

4 Query Version

The query version of the algorithm can be implemented by the same way as solving one instance of the problem. The preprocess is computing the all-pairs shortest path. Then we can answer any query including the starting point, target point, order of the polygons, and subset of polygons in $O(|E(WSP)| + kMWV \log V/\delta)$ time. First we will compute the distances from s to all points on P_1, then we will compute the distances from all points on last polygon to t. We have weight for other edges of WSP. Next we will use the dynamic algorithm mentioned earlier for calculation the shortest path. The number of edges from s to points in P_1 is at most kMW/δ. Calculating distance between s to any point takes $O(V \log V)$ time so it takes $O(VkMW \log V/\delta)$ to compute the distances to s. The same holds for t. The remaining of the algorithm takes $O(|E(WSP)|)$ time.

5 Other Applications of the Algorithm

5.1 TPP in Unweighted Regions

In this section we will apply our algorithm to unweighted version of the TPP. One of the strengths of our algorithm is that not only it works for concave polygon which can intersect with each other but also works for polygonal and non-closed shapes.

The general case of TPP can be solved in $O(|E(WSP)|)$ running time. We will put Steiner points on polygons with distance $\delta/2k$. Since the edges of WSP now correspond to Euclidean distance between points, computing the edges of WSP will take $O(1)$ running time with no error. The dynamic algorithm which mentioned earlier will find the answer, the running time will be $O(|E(WSP)|)$ which is $O(M^2k^3/\delta^2)$.

The error evaluation is similar to weighted problem. We will prove that there is a path π' in WSP which has distance at most $\pi^* + \delta$. The prove technique is the same as the weighted version. For each point p_i on π^*, there is point p'_i which has distance at most $\frac{\delta}{2k}$ from p_i. By using triangular inequality twice, we can show that the distance between p'_i and p'_{i+1} is at most $\frac{\delta}{k}$ plus distance between p_i and p_{i+1}. Starting from s, using p'_is, and ending at t, a path π' will be constructed which has distance at most δ from π^*. Having the fact that π is shortest path in WSP implies $\pi < \pi' < \pi^* + \delta$.

5.2 Applying the Algorithm for Other Problems

On some version of TPP, there is a constraint that the path from P_i to P_{i+1} should remain inside a fence. These version can be handled by performing a separate Dijkestra for each of the edges between P_i and P_{i+1} with edges out of fence removed. Another way is performing k times all-pairs shortest path where each time the edges are limited t by a particular fence.

Solving this version will help up solve other TPP problems in weighted regions including "Zoo Keepers", "Watchman Route" and "Safari". "Watchman Route" problem includes unweighted versions which edges of the pockets(Polygons) are curve(Some angular constraints usually cause these cases). As mentioned earlier, our algorithm works for cases were the inputs consists of polygonal objects, so our algorithm will solve these versions of "Watchman Route" problem.

6 Conclusion and Future works

Although the proposed algorithm has not polynomial running time in size of given input bits and depends on the lengths of edges, on the other hand it is the first result in "TPP in weighed regions". It is simple to implement. In practice for maps with long edges, big numbers for δ are acceptable, so the running time of the algorithm will not depend on the length of edges. Also we showed that our result is useful for special cases on unweighted regions.

Proposed algorithm's running times does not seem to be optimal. Other techniques mentioned on the introduction part of the paper maybe applicable on the problem to improve the running time and may result $(1+\epsilon)$-approximation algorithms. Algorithms independent of perimeter of the polygons are more applicative. Finding such algorithms is another way for improvement.

Other improvement is to limit the constraints to make a problem having exact solution with polynomial running time.

Summary of Results:

1. An approximation algorithm with $O(((n^3LW + RW)\frac{k}{\delta})^3)$ running time for the touring a sequence of polygons in weighted regions.
2. An approximation algorithm with $O(((n^3LW + RW)\frac{k}{\delta})^3)$ preprocess time and $O(M^2W^2k^3/\delta^2)$ query time for touring a sequence of polygons problem in weighted regions, where query can include start and target point, a subset of polygons and order of them.
3. An approximation algorithm with $O(M^2k^3/\delta^2)$ running time for the general case of touring a sequence of polygons.
4. Solving touring a sequence of polygons in weighted regions will result solving "Zoo Keepers", "Watchman Route" and "Safari" problems in weighted regions.

7 Acknowledgment

The authors gratefully acknowledge Ramtin Khosravi and Mojtaba Noori for their fruitful comments.

References

[1] J.S.B. Mitchell and C.H.Papadimitriou. "The weighted region problem: Finding shortest paths through a weighted planar subdivision." *journal of the ACM,* 38(1):18-73, Jan. 1991.

[2] C.Mata and J.Mitchell. "A new algorithm for computing shortest paths in weighted planar subdivisions".In *proceedings of the 13th Annual ACM Symposium on Computational Geometry,* pages 264-273, 1997.

[3] Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph S. B. Mitchell: "Touring a Sequence of Polygons", *Proc. 35th ACM Symposium on Theory of Computing (STOC 2003),* San Diego, CA, June 9-11, 2003, pp. 473–482.

[4] L.Aleksandrov, M.Lanthier,A.Maheshwari, and J.-R.Sack. "An ϵ-approximation algorithm

for weighted shortest paths on polyhedral surfaces". In *proceedings of the 6th Scandinavian Workshop on Algorithms Theory*, volume 1432 of *Lecture Notes in Computer Science*, pages 11-22 ,1998.

[5] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. "Approximation algorithms for geometric shortest path problems". In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 286-295, 2000.

[6] M. Lanthier, A. Maheshwari, and J.-R. Sack. "Approximating weighted shortest paths on polyhedral surfaces". In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 274–283, 1997.

[7] J.S.B. Mitchell. "Geometric shortest paths and network optimization". In J.-R.Sack and J. Urrutia, editors, *handbook of Computational Geometry*, pages 633-701. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[8] J. Reif and Z. Sun. "An efficient approximation algorithm for weighted region shortest path problem". In *Proceedings of the 4th Workshop on Algorithmic Foundations of Robotics*, 2000.

[9] J. Reif and Z. Sun. "On Finding Approximate Optimal Paths in Weighted Regions" unpublished.