Minimum weight pseudo-triangulations

Joachim Gudmundsson [a,1] Christos Levcopoulos [b]

[a] *Department of Mathematics and Computing Science, TU Eindhoven, 5600 MB, Eindhoven, the Netherlands.*
[b] *Department of Computer Science, Lund University, Box 118, 221 00 Lund, Sweden.*

Abstract

We consider the problem of computing a minimum weight pseudo-triangulation of a set S of n points in the plane. We first present an $O(n \log n)$-time algorithm that produces a pseudo-triangulation of weight $O(wt(M(S)) \cdot \log n)$ which is shown to be asymptotically worst-case optimal, i.e., there exists a point set S for which every pseudo-triangulation has weight $\Omega(\log n \cdot wt(M(S)))$, where $wt(M(S))$ is the weight of a minimum spanning tree of S. We also present a constant factor approximation algorithm running in cubic time. In the process we give an algorithm that produces a minimum weight pseudo-triangulation of a simple polygon.

Pseudo-triangulations are planar partitions that recently received considerable attention mainly due to their applications in visibility [7], ray-shooting [3], kinetic collision detection [4], rigidity [10], and guarding [9].

A pseudo-triangle is a planar polygon that has exactly three convex vertices, called corners, with internal angle less than π. A pseudo-triangulation of a set S of n points in the plane is a partition of the convex hull of S into pseudo-triangles whose vertex set is exactly S.

A related problem is the problem of triangulating a point set. Minimizing the total length has been one of the main optimality criteria for triangulations and other kinds of partition. The complexity of computing a minimum weight triangulation is one of the most longstanding open problems in computational geometry and it is included in Garey and Johnson's [1] list of problems from 1979 that neither are known to be NP-complete, nor known to be solvable in polynomial time. As a result approximation algorithms for the MWT-problem have been considered. In this paper we consider the problem of computing a pseudo-triangulation of minimum weight (MWPT) which was posed as an open problem by Rote et al. in [8]. An interesting observation that makes the pseudo-triangulation favorable compared to a standard triangulation is the fact that there exists point sets where any triangulation, and also any convex partition (without Steiner points), has weight $\Omega(n \cdot wt(M(S)))$, while there always exists a pseudo-triangulation of weight $O(\log n \cdot wt(M(S)))$, where $wt(M(S))$ is the weight of a minimum spanning tree of the point set. We also present an approximation algorithm that produces a pseudo-triangulation whose weight is within a factor 27 times the weight of the MWPT. In comparison, the best constant approximation factor for the MWT-problem, Levcopoulos and Krznaric [6], which is proved to be achievable by a polynomial-time algorithm [6] is so much larger that it has not been explicitly calculated.

An edge/segment with endpoints in two points u and v of S will be denoted by (u, v) and its length $|uv|$ is equal to the Euclidean distance between u and v. Given a graph T on S we denote by $wt(T)$ the sum of all the edge lengths of T. The minimum spanning tree of S and the convex hull of S, denoted $M(S)$ and $CH(S)$ respectively, will be used frequently throughout the paper. Both structures can be computed in $O(n \log n)$ time.

1. A fast pseudo-triangulation

As mentioned in the introduction there exist point sets S where any triangulation will have

Email addresses: `h.j.gudmundsson@tue.nl` (Joachim Gudmundsson), `christos@cs.lth.se` (Christos Levcopoulos).

[1] Supported by the Netherlands Organisation for Scientific Research (NWO).

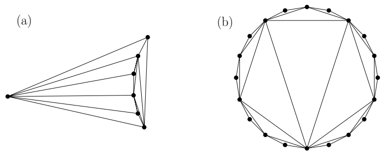

Fig. 1. (a) An example where any triangulation will have weight $\Omega(wt(M(S)) \cdot n)$. (b) An example where any pseudo-triangulation will have weight $\Omega(wt(M(S)) \cdot \log n)$.

weight $\Omega(n \cdot wt(M(S)))$, an example is given in Fig. 1a. A natural question is whether there exists similar worst-case bounds for pseudo-triangulations. In this section we show that one can always construct a pseudo-triangulation of weight $O(\log n \cdot wt(M(S)))$, and this is asymptotically tight, i.e., there exists a point set S for which every pseudo-triangulation has weight $\Omega(\log n \cdot wt(M(S)))$. We start with the lower bound.

Observation 1 *There exists a point set S in the plane such that any pseudo-triangulation has weight $\Omega(wt(M(S)) \cdot \log n)$.*

Next we present an algorithm that produces a pseudo-triangulation whose weight asymptotically meets the lower bound, that is:

Theorem 1 *Given a set S of n points in the plane one can in time $O(n \log n)$ produce a pseudo-triangulation of S of weight $O(wt(M(S)) \cdot \log n)$.*

The algorithm performs two main steps: first a partition of the convex hull of S into simple polygons $P_1, \ldots, P_m$ followed by a pseudo-triangulation of each polygon.

In this section we first show how a visibility polygon P can be pseudo-triangulated in time $O(n \log n)$ using edges of total weight $O(wt(P) \cdot \log n)$. We also show how to pseudo-triangulate a special polygon, called an hourglass polygon. Finally it is shown how one can construct a spanning graph of S that partitions the convex hull of S into empty polygons that either are visibility polygons, or hourglass polygons by using segments of small total weight. Combining these results gives us Theorem 1.

1.1. *Pseudo-triangulating a visibility polygon*

We will show that a weak visibility polygons whose visibility edge has two convex vertices easily can be pseudo-triangulated. This result will be used in the algorithm that pseudo-triangulates a visibility polygon. First a simple observation.

Observation 2 *The geodesic shortest path between any pair of points p and q in a weak visibility polygon P is a concave chain.*

Observation 3 *A weak visibility polygon P whose visibility edge (p_1, p_2) has two convex vertices can be pseudo-triangulated in time $O(n \log n)$ using edges of total weight $O(wt(P) \cdot \log n)$.*

Now we are ready to consider visibility polygons. Assume that we are given a visibility polygon P with respect to q with n vertices $p_1, \ldots, p_n$ ordered clockwise around the perimeter of P starting with q. Let $r_1, \ldots, r_m$ be the convex vertices of P.

Observation 2 implies that we can partition P into one pseudo-triangle and a set of weak-visibility polygons by adding the pseudo-triangle with corners at p_1, r_i and r_j, where $1 < i < j$. The two convex vertices r_i and r_j are chosen in such a way that the two angles $\angle p_2, p_1, r_i$ and $\angle p_n, p_1, r_j$ are less than π. Note also that p_2 and p_n are convex vertices since P is a visibility polygon. The pseudo-triangle will consist of the edges in the concave chain between r_i and r_j plus the edges (p_1, r_i) and (p_1, r_j). The resulting subpolygons outside the pseudo-triangle are weak visibility polygons whose visibility edges have convex vertices. According to Observation 3 each of these subpolygons can be pseudo-triangulated in $O(n \log n)$ time using edges of total weight $O(wt(P) \cdot \log n)$. Hence we have showed the following lemma.

Lemma 2 *The algorithm produces a pseudo-triangulation T of a visibility polygon P in $O(n \log n)$ time whose weight is $O(wt(P) \cdot \log n)$.*

We end this section by considering the pseudo-triangulation of an hourglass polygon. A polygon P is said to be an hourglass polygon if P consists of two concave chains connected by two edges.

We will later need the following straight-forward observation:

Observation 4 *An hourglass polygon P can be pseudo-triangulated in linear time by adding one edge e such that $wt(e) \leqslant 1/2 \cdot wt(P)$.*

1.2. *Partition a point set into simple polygons*

As input we are given a set S of n points in the plane, and as output we will produce a set of polygons that are either hourglass polygons or visibility polygons. The partition is done in two main steps. First construct the convex hull and the minimum spanning tree of S. This is done in $O(n \log n)$ time and it partitions CH(S) into simple (maybe degenerate) polygons, denoted $P_1, \ldots, P_m$. Secondly, each polygon P_i is processed independently. The task at hand is to partition P_i into a set of hourglass polygons and "restricted" visibility polygons, which can be pseudo-triangulated as described in the previous section.

A *restricted* visibility polygon $rvp(P, q)$ of a polygon P with respect to a vertex q is a visibility polygon of P with respect to q such that every vertex of $P(q)$ also is a vertex of P.

Definition 3 *Every edge $e = (u, v)$ of a restricted visibility polygon $R(q)$ that short cuts exactly three edges of the maximal visibility polygon $P(q)$ is said to be a split edge.*

Now, let $v_1, \ldots, v_n$ be the vertices of P in clockwise order, starting at $q = v_1$. It remains to show how we can partition P into visibility polygons and hourglass polygons in $O(n \log n)$ time. The idea is to recursively partition P into restricted visibility polygons and hourglass polygons. Consider one level of the recursion. If P is not a restricted visibility polygon with respect to q, or an hourglass polygon then the following two steps are performed:

(i) Build a restricted visibility polygon $rvp(P, q)$ of P.

(ii) For each split edge e in $rvp(P, q)$ construct an hourglass polygon H such that $H \cap R(q) = e$.

A more precise description on how this can be performed in time $O(n \log n)$ can be found in the full version.

2. A MWPT of a simple polygon

Even though the above algorithm is asymptotically worst-case optimal with respect to the weight of the minimum spanning tree it can be very far from the optimal solution. In the rest of this paper we will focus on developing a constant factor approximation algorithm for the MWPT-problem. As a subroutine we will also develop an algorithm that finds an optimal pseudo-triangulation of a simple polygon.

Theorem 4 *Given a simple polygon P one can compute the minimum weight pseudo-triangulation of P in $O(n^3)$ time using $O(n^2)$ space.*

We will use a similar dynamic programming method as proposed by Gilbert [2] and Klincsek [5] for finding a minimum weight triangulation of a simple polygon. The basic observation used is that once some (pseudo-)triangle of the (pseudo-)triangulation has been fixed the problem splits into subproblems whose solutions can be found recursively, hence avoiding recomputation of common subproblems.

Let P be the simple polygon with n vertices $p_1, \ldots, p_n$ in clockwise order. Let $\delta(p_i, p_{i+j})$ be the shortest geodesic path between p_i and p_{i+j}. Define the *order* of a pair of points p_i, p_j to be the value $(i - j - 1) \mod n$, i.e., the number of vertices on the path from p_i to p_j along P in clockwise order. Sort the pairs with respect on their order, ties are broken arbitrarily. Note that every pair of points p_i and p_j will occur twice; once as (p_i, p_j) and once as (p_j, p_i). Now we process each pair in sorted order as follows.

Assume we are about to process (p_i, p_{i+j}) and that the path $\delta(p_i, p_{i+j})$ goes through the vertices $p_i = p_{i+a_0}, p_{i+a_1}, \ldots, p_{i+a_k} = p_{i+j}$. Note that the path partitions P into $k + 1$ subpolygons. Let $L[i, i + j]$ be the total edge length of an optimal pseudo-triangulation for the subpolygon (or subpolygons) containing the chain $p_i, p_{i+1}, \ldots, p_{i+j}$ of the perimeter of P. Compute $L[i, i + j]$ recursively as follows. If (p_i, p_{i+j}) is not a convex or concave chain then we set $L[i, i + j] = \infty$. In the case when the path is a concave or convex chain we obtain one polygon P' bounded by the path $\delta(p_i, p_{i+j})$ and the path between p_i and p_{i+j}, and k polygons $P_1, \ldots, P_k$ where each P_l is bounded by the edge $(p_{i+a_l}, p_{i+a_{l-1}})$ and the edges from $p_{i+a_{l-1}}$ to p_{i+a_l} along the perimeter of P. If the path is a concave or convex chain then we will have three cases.

- If $\delta(p_i, p_{i+j})$ contains more than one edge then we know that $L[*, *]$ already has been computed for every edge along $\delta(p_i, p_{i+j})$, hence we only have to add up the values of $L[*, *]$ which can be done in linear time, i.e., calculating $\sum_{\alpha=0}^{k-1} L[p_{i+a_\alpha}, p_{i+a_{\alpha+1}}]$.

- If $\delta(p_i, p_{i+j})$ contains exactly one edge (p_i, p_{i+j}) then an optimal pseudo-triangulation of P_1 can be obtained in linear time as follows. We will have two cases; either p_i and p_{i+j} are corners of the pseudo-triangle in P_1 containing (p_i, p_{i+j}) or not.

In the case when both p_i and p_{i+j} are convex vertices within P_1 then an optimal pseudo-triangulation of P_1 can be obtained in linear time as follows. Any optimal pseudo-triangulation of P_1 that contains the edge (p_i, p_{i+j}) must have p_i and p_{i+j} as corners thus we can try all possible vertices p_m, $i < m < i + j$ as the third corner.

Testing a pseudo-triangle with corners at p_i, p_{i+j} and p_m takes constant time since the $L[*,*]$-value of the paths between p_i and p_m, and p_m and p_{i+j} already has been computed.

Otherwise, if one or both of the points are not corners, it holds that there must be a pair of points p_x and p_y along the perimeter of P between p_i and p_{i+j} whose shortest geodesic path between them contains the edge (p_i, p_{i+j}). Hence, in this case the optimal solution has already been computed for P_1.

There are $O(n^2)$ pairs of points and each pair takes $O(n)$ time to process. The space bound follows from the fact that for every pair of points p_i and p_j we store $L[p_i, p_j]$. When all the $L[*,*]$ have been computed we can easily test every possible pseudo-triangle in constant time, thus Lemma 4 follows.

3. A better approximation

In this section we will give an approximation algorithm for the MWPT-problem. It is similar to the approximation algorithm presented in Section 1 in the sense that the two main steps are the same; first a partition of the convex hull of the point set into simple polygons followed by a pseudo-triangulation of each polygon. In the pseudo-triangulation step we will use the optimal algorithm presented in the previous section. As input we are given a set S of n points in the plane, and as output we will produce a pseudo-triangulation T of S.

Algorithm PSEUDOTRIANGULATE(S)

(i) Construct the convex hull and the minimum spanning tree of S. This partitions $CH(S)$ into simple (maybe degenerate) polygons denoted $Q_1, \ldots, Q_k$.

(ii) Apply Theorem 4 to each of the k polygons. The pseudo-triangulation obtained together with the convex hull and the minimum spanning tree of S is reported.

The proof of the following theorem can be found in the full version of the paper.

Theorem 5 *Given a set of points S algorithm* PSEUDOTRIANGULATE *computes a pseudo-triangulation T of S in time $O(n^3)$ using $O(n^2)$ space such that $wt(T) = 4(1+4\sqrt{2}) \cdot wt(T_{opt})$, where T_{opt} is a minimum weight pseudo-triangulation of S.*

4. Open problems and Acknowledgement

An obvious question is whether the minimum weight pseudo-triangulation problem is as hard as finding the minimum weight triangulation? The MWT-problem is one of the few open problems listed in Garey and Johnson's 1979 book on NP-completeness [1] that remain open today.

A second open problem concerning the weight of a pseudo-triangulation is if there exists a minimum pseudo-triangulation of low weight. It was shown by Streinu [10] that every point set allows a minimum planar pseudo-triangulation that has $2n-3$ edges. Neither of the two algorithms presented in this paper produces minimum pseudo-triangulations, although the dynamic programming algorithm for simple polygons can be modified to compute a minimum weight minimum pseudo-triangulation.

The first author would like to thank Mark de Berg and Bettina Speckmann for valuable discussions during the work on this paper.

References

[1] M. Garey and D. Johnson. Computers and Intractability. W. H. Freeman and Company, 1979.

[2] P. D. Gilbert. New results in planar triangulations. Report R–850, Univ. Illinois Coord. Science Lab, 1979.

[3] M. T. Goodrich and R. Tamassia. Dynamic Ray Shooting and Shortest Paths in Planar Subdivisions via Balanced Geodesic Triangulations. Journal of Algorithms, 23(1):51–73, 1997.

[4] D. Kirkpatrick and B. Speckmann. Kinetic Maintenance of Context-Sensitive Hierarchical Representations for Disjoint Simple Polygons. Proc. 18th ACM Symp. on Computational Geometry, 2002.

[5] G. Klincsek. Minimal triangulations of polygonal domains. Annals of Discrete Math., 9:121-123, 1980.

[6] D. Krznaric and C. Levcopoulos. Quasi-Greedy Triangulations Approximating the Minimum Weight Triangulation. J. of Algorithms 27(2): 303-338. 1998.

[7] M. Pocchiola and G. Vegter. Pseudo-triangulations: Theory and applications. In Proc. 12th ACM Symposium on Computational Geometry, 1996.

[8] G. Rote, C. A. Wang, L. Wang, and Y. Xu. On constrained minimum pseudotriangulations. Proc. 9th Symposium on Computing an Combinatorics, 2003.

[9] B. Speckmann and C. D. Tóth. Allocating Vertex pi-guards in Simple Polygons via Pseudo-Triangulations. Proc. 14th ACM-SIAM Symp. on Discrete Alg., 2003.

[10] I. Streinu. A Combinatorial Approach to Planar Non-Colliding Robot Arm Motion Planning. Proc. 41st ACM Symposium on Found. of Comp. Sci., 2000.