MINIMAL TRIANGULATIONS OF POLYGONAL DOMAINS

G.T. KLINCSEK

School of Computer Science, McGill University, 805 Sherbrooke St. W., Montreal, Que. H3G 2K5, Canada

1. Introduction

Let V be a set of n distinct points (vertices) $M_1, M_2, \ldots, M_n$ in the plane. We assume that no 3 points are collinear. This assumption is not essential (as long as not all the points are collinear); but simplifies the explanations. Let E be the family of $\frac{1}{2}n(n-1)$ line segments (edges) joining the vertices of V.

Definition. A *triangulation T of V* is a maximal subset of E in which no two edges cross each other.

Clearly, in the planar graph determined by V and T each interior face is a triangle.

The weight $s(T)$ of a triangulation T is the sum of the length of the edges in T.

Definition. The *minimal weight triangulation* MWT is a triangulation on V for which $s(T)$ is minimal.

Let M.W.T. also denote the weight of this triangulation.

The problem of finding MWT presents many intriguing aspects:

(a) Several very fast algorithms were proposed and later proven wrong. For example, the Delaunay triangulation (which can be obtained in $n \log n$ time) [3] does not always give the exact answer [1].

(b) For the heuristics in use, little information is available about their error [2].

(c) Better analyzed problems, like the minimal-spanning-tree and the minimal-Hamiltonian circuit problems do not give helpful information. There are examples to show that the MWT needs not to contain the minimal spanning tree or any Hamiltonian circuit [1].

(d) There is some evidence to suspect that the problem is NP-complete, but no proof to date is available [1].

To gain some insight of the general problem this paper proposes to solve a variant of the intitial problem, as described in the next paragraph.

2. Restricted minimal triangulations

Let S be a given subset of E where no two edges of S cross each other. Then there exists some triangulation T such that $S \subset T$.

Definition. The restricted triangulation problem consists of finding a T of minimal weight among those containing S.

If S is a connected spanning graph over the vertices of V, the problem is solved using Algorithm B of Section 4.

Since any triangulation contains the convex hull C^0 of the graph (V, E), we can start with $S^0 = S \cup C^0$. And hence the initial condition can be relaxed as: let $S \cup C^0$ be connected and spanning. As S^0 separates the plane into a number of connected regions, we can apply Algorithm B to each simple domain and our answer is the union of the individual triangulations.

Hence we have a rule of thumb to improve a given triangulation: select a set of n edges of the given triangulation. These should contain all the edges of C^0 and span V. In other words, grow a "spanning" tree from the convex hull to the interior of V. A simple polynomial domain results, for which the best triangulation can be obtained in $O(n^3)$ operations. The question of the selection of the most appropriate spanning tree is still open, since this of course, would solve the general problem.

3. Algorithm A—Triangulation of a convex polygon

Let $M_1, M_2, \ldots, M_n$ be the vertices (ordered clockwise) of a convex polygon in the plane. To emphasize this ordering we will use the name M_{n+i} for the node M_i whenever M_i is reached "the second time around the perimeter".

The M.W.T. can be obtained using dynamic programming.

Let $C(i, j)$, where $(i<j)$, be the M.W.T. of the subgraph involving the nodes $M_i, M_{i+1}, \ldots, M_j$. Intuitively speaking, we cut off an area of the polygon along the segment $\overline{M_iM_j}$ and compute the M.W.T. of this piece.

Algorithm A

Step 1: For $k=1, i=1, 2, \ldots, n$ and $j=i+k$ let $C(i, j)=d(M_i, M_j)$, where $d(M_i, M_j)$ is the length of the segment M_iM_j.

Step 2: Let $k=k+1$. For $i=1, 2, \ldots, n$ and $j=i+k$ let

$$(*) \qquad C(i, j)=d(M_i, M_j)+\min_{i<m<j}[C(i, m)+C(m, j)].$$

For each pair (i, j) let $l=L(i, j)$ be the index where the minimum $C(i, j)$ in $(*)$ is achieved.

Step 3: If $k < n$ go to Step 2, otherwise the weight of M.W.T. is $C(1, n)$.

Step 4: To find the edges involved in the M.W.T. we should backtrace along the pointers L.

The edge $\overline{M_1M_n}$ is in M.W.T.

Step 5: For each $\overline{M_iM_j} \in$ M.W.T. with $j > i+1$ let $l = L(i, j)$, then $\overline{M_iM_l} \in$ M.W.T. and $\overline{M_lM_j} \in$ M.W.T.

4. Triangulation of a simple polygon *P* domain. Algorithm B

Let $M_1, M_2, \ldots, M_n$ be the vertices of a simple polygonal domain D. the vertices are numbered sequentially along the boundary. (By simple we mean that D is simply connected.) As before we introduce the names M_{n+i}.

For each segment M_iM_j $(j > i+1)$ we need a decision. M_iM_j is *interior* to D if the line-segment $\overline{M_iM_j}$ (not the straight line through M_i and M_j) divides D in exactly 2 components.

To find the M.W.T. for the interior of D we modify the distance function as follows:

$$d^*(M_i, M_j) = \begin{cases} d(M_i, M_j) & \text{if } \; j = i+1 \text{ or if } M_iM_j \text{ is interior to } D. \\ -\infty & \text{otherwise.} \end{cases}$$

Algorithm B is the same as Algorithm A but substitute d^* for d in Step 2.

5. Running time analysis

In Algorithm A the longest executing stage is Step 2. It requires a constant multiple of $n \times k$ operations for each $k = 1, 2, \ldots n$. Hence the total running time of the algorithm is of order n^3.

In Algorithm B the evaluation of d^* which may be done parallel to Step 2 or in a set-up stage, needs again at most $O(n^3)$ operations.

For each of the $n \times \frac{1}{2}(n-1)$ segments M_iM_j test for intersection with the $O(N)$ edges of the polygonial domain D. If there is intersection, M_iM_j is not interior to D. Otherwise M_iM_j is interior to D provided it is interior to the angle $M_{i-1}M_iM_{i+1}$.

References

[1] E.L. Lloyd. On triangulations of a set of points in the plane. Proc. 18th Annu. Symp. on Foundations of Computer Science. IEEE 228–240.

[2] G.K. Manacher and A.L. Zobrist. Neither the Greedy nor the Delaunay Triangulation of a planar point set approximates the optimal triangulation. Information Processing Letters 9 (1979) 31–34.

[3] M.I. Shamos. Geometric complexity. Proceedings of the Seventh ACM Symposium on the Theory of Computing (1975) 224–233.