An $O(n \log n)$ Heuristic for Steiner Minimal Tree Problems on the Euclidean Metric

J. MacGregor Smith
Department of Industrial Engineering and Operations Research, University of Massachusetts, Amherst, Massachusetts 01003

D. T. Lee
Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois 60201

Judith S. Liebman
Operations Research Laboratory, Department of Mechanical and Industrial Engineering, University of Illinois, Urbana, Illinois 61801

An $O(n \log n)$ heuristic for the Euclidean Steiner Minimal Tree (ESMT) problem is presented. The algorithm is based on a decomposition approach which first partitions the vertex set into triangles via the Delaunay triangulation, then "recomposes" the suboptimal Steiner Minimal Tree (SMT) according to the Voronoi diagram and Minimum Spanning Tree (MST) of the point set. The ESMT algorithm was implemented in FORTRAN-IV and tested on a number of randomly generated point sets in the plane drawn from a uniform distribution. Comparison of the $O(n \log n)$ algorithm with an $O(n^4)$ algorithm clearly indicates that the $O(n \log n)$ algorithm is as good as the previous $O(n^4)$ algorithm in achieving reductions in the ratio SMT/MST of the given vertex set. This is somewhat surprising since the $O(n^4)$ algorithm considers more potential Steiner points and alternative tree configurations.

I. THE EUCLIDEAN STEINER MINIMAL TREE PROBLEM

The Euclidean Steiner Minimal Tree (ESMT) problem is the Steiner network problem which has classically received the most attention in the literature [2, 5, 8, 10, 13, 14, 17-20]. The ESMT problem is as follows: for a given set V of points in the plane, where $V = \{v_1, v_{2_2} \dots, v_n\}$, we wish to construct a minimal length tree which connects these vertices. In order to minimize this total length, additional vertices $S = \{s_1, s_2, \dots, s_m\}$ are sometimes necessary. In this problem, the distance metric is the L_2 metric function, better known as the Euclidean metric.

Algebraically, we can represent the ESMT problem as follows:

Networks, Vol. 11 (1981) 23-39
© 1981 John Wiley & Sons, Inc.
CCC 0028-3045/81/010023-17$1.70

$$\text{Minimize } Z = \sum_{i \in S} \sum_{\substack{j \in V \cup S \\ i \neq j}} [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2},$$

since the objective function is strictly convex, the necessary and sufficient conditions for our solution to achieve a minimum are given by:

$$\partial Z/\partial x_i = \partial Z/\partial y_i = 0$$

In certain configurations, the location of Steiner points coincides with those of the given points.

Some properties of the optimal solution to this problem are known to be the following:

(1) The number of Steiner points is m, where $0 \leqslant m \leqslant n - 2$.

(2) The graph is planar; i.e., there exist no crossings in the network except at the nodes of the network.

(3) Each original vertex v_i, $1 \leqslant i \leqslant n$, has at most degree $d(v_i) \leqslant 3$.

(4) Each Steiner node s_j, $1 \leqslant j \leqslant m$, has degree $d(s_j) = 3$.

(5) It has been conjectured that the Minimum Spanning Tree (MST) is no more than $2/\sqrt{3}$ times as long as the minimum length ESMT [8].

Polak has most recently shown for $n = 4$ that this conjecture is true, however, for the general case $n \geqslant 4$, he does not see any foreseeable reasonable proof [21].

Graham and Hwang showed that the lower bound on the ratio of the SMT/MST $\geqslant 1/\sqrt{3}$ [9], and recently Chung and Hwang [3] have improved this bound to

$$\text{SMT/MST} \geqslant \frac{2\sqrt{3} + 2 - \sqrt{7 + 2\sqrt{3}}}{3}.$$

(6) The problem has recently been shown to be NP-complete [7].

II. OVERALL STRATEGY FOR THE $O(n \log n)$ ALGORITHM

The overall strategy for constructing the $O(n \log n)$ heuristic algorithm is a decomposition approach which incorporates two phases: (1) a reduction phase; and (2) an expansion phase.

Initially, the point set V is triangulated. Within each triangle, the local optimal solutions, if they exist, are found. During the second phase, the solutions for each triangle are reconfigured into a solution for the entire point set. The triangulation of the point set is the Delaunay triangulation, which is unique for a particular metric, and is therefore a metrical triangulation of the point set [16]. The second phase of the algorithm is carried out through the properties of the Voronoi diagram and Minimum Spanning Tree (MST) of the point set. Although our approach may seem similar to Karp's in his probabilistic partitioning approach for the traveling salesman problem, our partitioning approach is vastly different and our overall use of the Delaunay triangulation and Voronoi diagram to recompose a suboptimal Steiner Minimal Tree solution is in no way similar to his algorithm [12].

III. $O(n \log n)$ PRELIMINARIES

The Voronoi diagram has been primarily used in solving nearest neighbor problems [15, 22]. Figure 1 illustrates a Voronoi polygon and a Voronoi diagram for a set of ten points. Figure 2 illustrates the dual graph of the Voronoi diagram which is a triangulation, called the Delaunay triangulation. The advantage of this triangulation for the SMT problem is that each Delaunay triangle tends to be more nearly an equilateral triangle. This property is quite important in SMT problems since larger percentage

Shamos has shown that the Delaunay triangulation can be found in $O(n \log n)$ time in the worst case [23]. Also, Shamos has proved that the Voronoi diagram can be used to generate a MST of a completely connected graph in $O(n \log n)$ time. Hwang has recently shown that an $O(n \log n)$ algorithm is possible for the MST on the rectilinear metric by using the Voronoi diagram [10]. The reason that the Delaunay triangulation can be used interchangeably between metrics is because the Voronoi diagram has been most recently shown to be generalizable to any L_p metric [6, 16]. First some useful concepts and definitions are in order.

Perpendicular bisector. If we have two points, v_i and v_j, the locus of points closer to v_i than to v_j is defined by the half-plane delineated by the perpendicular bisector of

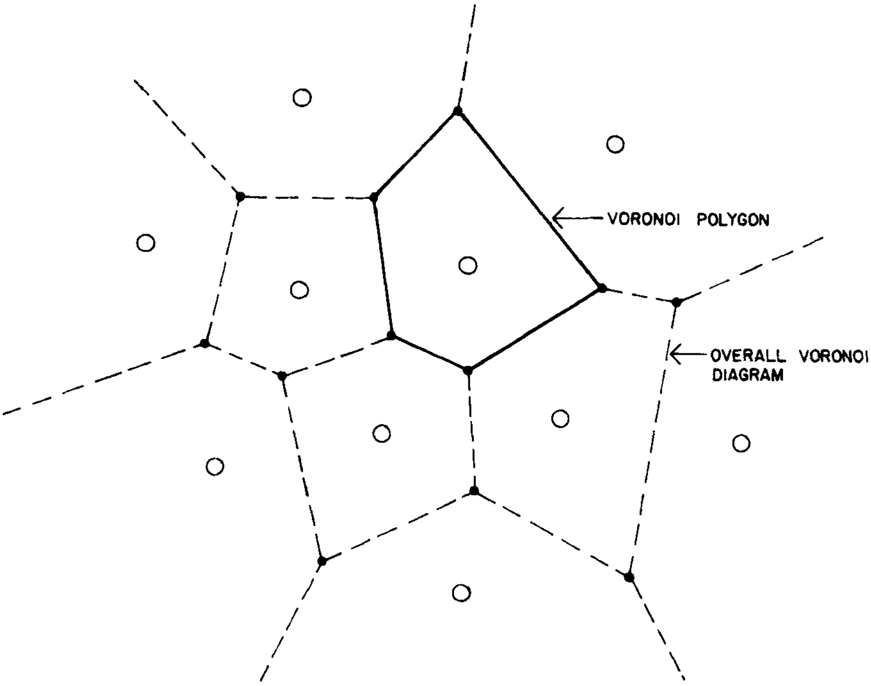


FIG. 1. Voronoi polygon and diagram.

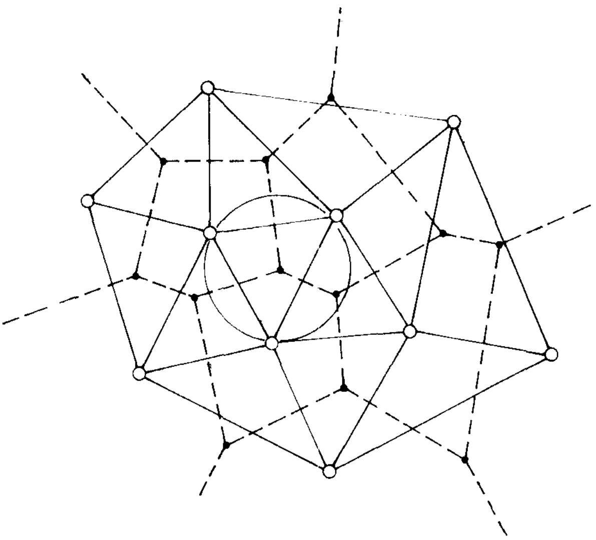


FIG. 2. Delaunay triangulation.

the line segment between v_i and v_j. The perpendicular bisector is defined as follows:

$$B_2(v_i, v_j) = \{r \in R_2^2 \mid d_2(r, v_i) = d_2(r, v_j)\}.$$

Voronoi polygon. A polygonal region that encompasses the locus of points closer to v_i than to any other point in V. Formally defined as the intersection of half-planes determined by the bisectors of v_i and all other points in V; often, denoted as VP(v_i).

$$\mathrm{VP}(v_i) = \bigcap_{i \neq j}^{n} H(v_i, v_j).$$

Voronoi diagram. The collection of Voronoi polygons VP(v_i) for each v_i in V. Often denoted as VD(V).

Voronoi point. Each vertex in the Voronoi diagram is a Voronoi point. Each Voronoi point is the circumcenter of at most three given vertices in R^2. The degenerate case with four or more cocircular points is excluded.

Delaunay triangulation. The planar straight line dual graph of the Voronoi diagram is a triangulation, often referred to as the Delaunay triangulation, denoted as DT(V).

There are some key properties of the Voronoi diagram and its dual, the Delaunay triangulation, which we should examine before delving into the details of the ESMT algorithm. Proofs of the validity of these lemmas occur in the cited references.

Lemma 1 [15, 23]. Each Voronoi point is the intersection of perpendicular bisectors of the line segments between pairs of points (v_i, v_j) in the plane. As such, each Voronoi point is the circumcenter of a triangle in the Delaunay triangulation.

Lemma 2 [15, 23]. Every circumcircle with the Voronoi point as its center contains no other vertex v_i of V in its interior.

Lemma 3 [15]. In the Voronoi diagram with $n > 3$ points:

$$E = 3(n - 1) - C,$$
$$I = 2(n - 1) - C,$$

where E = the number of edges in the Delaunay triangulation; I = the number of triangles; and C = the number of vertices on the perimeter of the Delaunay triangulation of the point set V.

Lemma 4 [23]. Any minimum spanning tree of the point set V is a subgraph of the Delaunay triangulation.

Lemma 4 follows from Lemma 3 and the basic properties of minimum spanning tree algorithms first proved by Kruskal and Prim [1]. This property is important when we construct the Steiner tree, since we will only have to consider $[3(n - 1) - C]$ edges, not the usual $[n(n - 1)]$ edges which would lead to an $O(n^2)$ algorithm.

Taken together, Lemmas 3 and 4 will lead to our desired $O(n \log n)$ ESMT algorithm. Like the Voronoi diagram, the Delaunay triangulation is constructed using a divide-and-conquer strategy, although implementation of the algorithm is done iteratively, and not recursively.

The Delaunay triangulation requires n circular doubly linked lists, one for each v_i in V. The edges are ordered within each list according to the polar angle of the edge at its initial vertex. While the details of the algorithm are described elsewhere [25], it is important to realize that the data structure established for the Delaunay triangulation carries over into the expansion phase for constructing the solution for the ESMT problem. It is this data structure which largely allows one an $O(n)$ step in constructing the ESMT once the triangulation is done.

IV. CONCATENATION ALGORITHM OVERVIEW

The algorithm for constructing the Steiner tree based upon the Delaunay triangulation operates in a manner similar to Kruskal's mininum spanning tree algorithm. However, in this case we establish a priority queue of triangles, based on their SMT/MST reductions, and then systematically construct a suboptimal SMT solution by a fast disjoint-set-union procedure. The reasons for this is that each triangle has a potential disjoint SMT in the forest of SMT's of V. In order to clarify notation, a suboptimal SMT will be indicated by $\widehat{\text{SMT}}$. Constructing the Delaunay triangulation assists us in defining the triangles, and information from the Voronoi diagram assists us in defining how to concatenate these triangles into a $\widehat{\mathrm{SMT}}$ for V. In carrying out this process, we establish a framework around which either an exact or near optimal Steiner point can be found for a triangle which has the potential to reduce the overall length of the $\widehat{\mathrm{SMT}}$. Once those triangles with the largest SMT/MST reductions have been identified and their Steiner points located, the concatenation process can be done in *linear* time.

V. ALGORITHM PRELIMINARIES

The Delaunay triangulation, by itself, does not provide us enough information for determining which are the best triangles to concatenate. Since there are a total of $2(n - 1) - C$ triangles, we must determine a proper subset of these to concatenate.

An important insight of Gilber and Polak in their classic paper [8], was that the MST could act as a guide in constructing Steiner minimal trees. Use of the MST as a guide is not useful in only certain instances and usually these cases are not drawn from a set of uniformly distributed points in the plane. An example of an instance where the MST is not a good guide for the SMT is pointed out by Chang [2], who also first implemented the MST as a guide in constructing the solution for the ESMT problem. Chang's algorithm was discussed previously [24]. We also will utilize the MST to indicate which is the proper subset of triangles to concatenate for the generalized $\widehat{\mathrm{SMT}}$ algorithm. Before we provide the details of the algorithm, however, we will establish some lemmas crucial to proving that we can achieve an $O(n \log n)$ algorithm for the generalized $\widehat{\mathrm{SMT}}$ problem. These lemmas now follow:

Lemma 5 [23]. The Delaunay triangulation affords an $O(n \log n)$ algorithm for the MST problem.

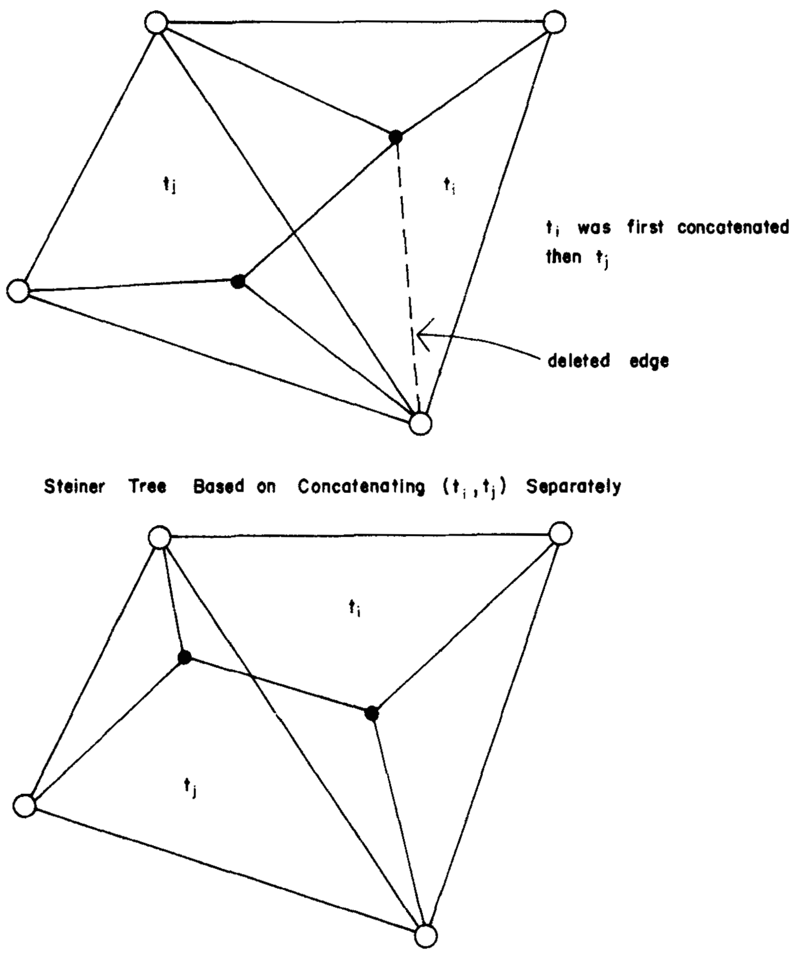
As we shall see, constructing the MST of V greatly assists us in identifying the candidates of the Delaunay triangles for reducing the overall MST length of V, resulting in a $\widehat{\mathrm{SMT}}$ of V. Additionally, utilizing the MST as the indicator of which triangles to concatenate guarantees an upper bound on the performance of the $\widehat{\mathrm{SMT}}$ algorithm. The upper bound is, of course, the length of the MST itself.

Lemma 6. For any given triangle t_i in the Delaunay triangulation with two of its edges in the MST of V, a Steiner point s_i can be inserted into t_i to reduce the overall MST of V if it has a SMT/MST < 1.00 and if the insertion does not create a cycle in V.

Proof: In constructing the MST of V, each triangle that has two of its edges in the MST is a potential candidate for a Steiner insertion, since the vertices of t_i, (v_i, v_j, v_k), are locally connected within the triangle. This is true no matter which L_p metric we utilize, as long as $1 \leqslant p \leqslant 2$. An additional vertex, s_i only can be inserted into t_i when the ratio of the SMT/MST is less than 1.00. Otherwise, insertion would increase the length of the MST of t_i and the overall MST of V. Finally, if inserting the s_i and the necessary edges connecting the vertices of t_i does not form a cycle in V, then we should go ahead and insert s_i deleting the necessary MST edges and adding the appropriate SMT edges. Q.E.D.

The ratio decision rule SMT/MST of Lemma 6 affords us a means of also identifying the order in which to consider the triangles for possible Steiner insertions. Further, to guarantee that no cycles are created, we will employ a fast find and disjoint-set-union procedure to determine whether the vertices in the S$\hat{M}$T are in the same disjoint tree or not [1].

Following many experiments with constructing Steiner minimal trees on the L_2 metric, it became apparent that certain configurations of points indicated that pairs of triangles with a total of four vertices could be concatenated simultaneously rather than as separate triangles of three vertices each. By selecting appropriate pairs of triangles, it was possible to achieve better solutions by concatenating four vertices at a time, especially on the L_p metric, where $1 \leqslant p \leqslant 2$. In terms of balancing speed against efficacy of the heuristic, Hwang argues that it is practical to consider adding three vertices at a time during the concatenation period. However, his experience has been primarily with S$\hat{M}$T's on the L_1 metric [10]. Results from many experiments indicate that for the L_2 metric, concatenating four vertices at a time is clearly superior to concatenating three at a time. Figure 3 illustrates a case in point. In fact, unless we



Steiner Tree Based on Concatenating (t_i, t_j) Separately

Steiner Tree Based on Concatenating (t_i, t_j) Together

FIG. 3. Comparison of concatenation procedures.

concatenate four vertices together we will not achieve the optimal solution for the four vertices in this example.

Because of the potential for obtaining better solutions, we need a rule to indicate which four-point clusters, i.e., which triangle pairs (t_i, t_j) of the Delaunay triangulation, should be concatenated.

Before presenting the rule, however, the following definitions and discussion are relevant.

Convex quadrilateral. This is a figure which has four points in the plane, no three of which are collinear. In addition, all four points are interconnected by four nonintersecting edges, and each point subtends an interior angle less than 180°.

Polak demonstrated that if a Full Steiner Tree (FST), one with $n - 2$ Steiner points, exists for a quadrilateral in the plane, the quadrilateral must be convex [21]. Further, he showed that for four points in the plane, a FST, if it exists, will approach the maximum possible reduction $\sqrt{3}/2$ in the ratio of the SMT/MST.

Regular convex quadrilateral. This is a convex quadrilateral with four equal sides and four equal angles, i.e., a square. An important property of a square is that the Voronoi points of the two adjacent Delaunay triangles of the square are coincident.

Rule I. Given a Delaunay triangle t_i, the most regular convex quadrilateral among the three quadrilaterals formed by t_i and its neighboring triangles is the ordered pair (t_i, t_j) which is convex and for which $\|vp(t_i) - vp(t_j)\|$ is a minimum, where $vp(t_i)$ is the circumcenter of triangle t_i.

From Lemma 2 we know that $vp(t_i)$ contains no other vertex v in its interior. In the degenerate case, however, $vp(t_i)$ could be the circumcenter of four points. For the degenerate case, we have two coincident Voronoi points for which $\|vp(t_i) - vp(t_j)\| = 0$. Ignoring this degenerate case for the moment, the degree of $vp(t_i)$ is equal to three. Since there are at most three neighboring triangles to t_i with Voronoi edge lengths > 0, the minimum Voronoi edge length of the three neighboring triangles will determine the most regular convex quadrilateral adjacent to t_i, if it exists.

Three comments are important at this juncture. First of all, it should not be inferred from the above that the unit square achieves a maximum SMT/MST reduction ratio of $\sqrt{3}/2$. However, from empirical results, this Rule has led to very good reductions. What is important about the above Rule is the convexity property it demonstrates for four points which can be directly inferred from the Voronoi diagram. As was mentioned previously, the convexity property is crucial to the existence of a FST which in turn is essential to achieving the maximum reduction SMT/MST for four points. Second, computational experience with generating Steiner trees for points in the plane has led us to believe that as you increase the number of Steiner points in the $\widehat{\text{SMT}}$, you tend to increase the reduction ratio $\widehat{\text{SMT}}$/MST. Although this is not always the case, it appears to be true for convex sets of points, i.e., with no interior points. Thus, use of the Voronoi diagram and Delaunay triangulation allows one to locate convex sets of points for which the FST, if it exists, can be generated for the component. Finally, for point clusters of more than four points, we conjecture and do not prove that FST's providing the maximum reduction in $\hat{\text{SMT}}$/MST only exist for convex point clusters. Example FST point configurations for $n > 4$ appear in Gilbert and Polak's paper [8]. Hence, if we have an efficient way of locating convex point clusters in the plane, we should be able to generate suitably large $\hat{\text{SMT}}$/MST reductions for the entire point set. This leads naturally to the following definitions and main heuristic rule:

Regular Convex n-gons. These figures have n equal sides and n equal interior angles, e.g. for $n = 5$, a pentagon; for $n = 6$ a hexagon, and so on. Additionally, the Delaunay triangles of the convex n-gons all have coincident Voronoi points.

Voronoi Tree. This is a subgraph of the Voronoi diagram which has no cycles and has exactly $n_p - 1$ edges, where n_p is the number of Voronoi points in the tree.

Minimum Spanning Voronoi Tree (MSVT). A tree spanning n_p Voronoi points with minimum length.

Rule II. Given a Delaunay triangle, t_i, the most regular convex n-gon among the n-gons formed by t_i and its neighboring triangles is the set of triangles $\{t_i, \ldots, t_k\}$ whose boundary is convex and whose MSVT is a minimum.

From Rule I, for $k = 2$, we know that the most regular convex quadrilateral, if it exists, is given by the minimum length Voronoi edge incident to the Voronoi point of t_i. The MSVT of $k = 2$ is thus a single edge of the Voronoi diagram for V.

The addition of a minimum weight Voronoi edge to the existing set of minimum weight Voronoi edge segments increases the number of unique vertices in the point cluster by one and the number of triangles by one. If the additional edge added to the existing MSVT did not add a unique vertex to the existing cluster of vertices, then it would necessarily be an edge which would form a cycle in the MSVT. Forming a cycle in the MSVT is equivalent to connecting two triangles already included in the construction of the MSVT. Thus, if the addition of a minimum weight Voronoi edge adds a unique vertex to k it must also add a new triangle to the cluster of triangles, otherwise, the addition of the Voronoi edge would create a cycle in MSVT. Further, since the edge added to the MSVT is the minimum edge incident to the Voronoi vertices already in MSVT, the unique vertex should be the vertex which creates the most regular convex collection of points, if it exists, of the k adjacent Delaunay triangles. Figure 4 illustrates the process.

The question naturally arises through Rule II, whether concatenating five or six vertices at a time might lead to even better $\hat{\text{SMT}}$ solutions. This would be done by first defining the nearest pair of triangles (t_i, t_j) as in Rule I, then determining the shortest Voronoi edges incident to the Voronoi edge of the triangle pair (t_i, t_j). For five vertices, the nearest triple of triangles, starting with t_i would be found. Going one step further, we could find the nearest neighbor cluster of four triangles which would result in finding the most regular convex cluster of six vertices; see Figure 5. With these clusters of five and six vertices, we could then employ an exact or heuristic method for determining the Steiner tree for the 5 and 6 vertex configurations found

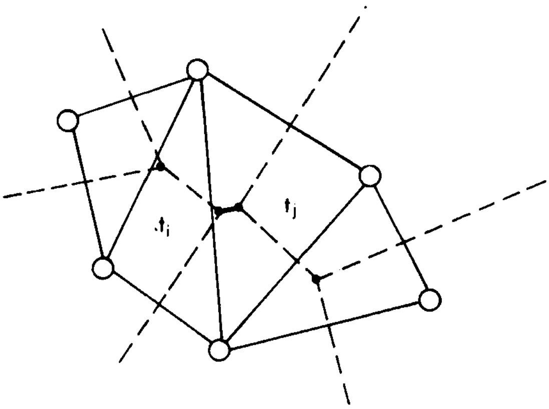


FIG. 4. A nearest neighbor triangle pair (t_i, t_j).

through the use of the Voronoi edge information. However, preliminary experiments with concatenating five vertices at a time has indicated that this is not necessarily a better strategy than alternating between concatenating three and four vertices at a time. These experimental results are based primarily on the use of the algorithm on point sets drawn from a uniformly distributed set of points in E^2 bounded by a unit square. For other point sets drawn from different distributions, this may not be the case.

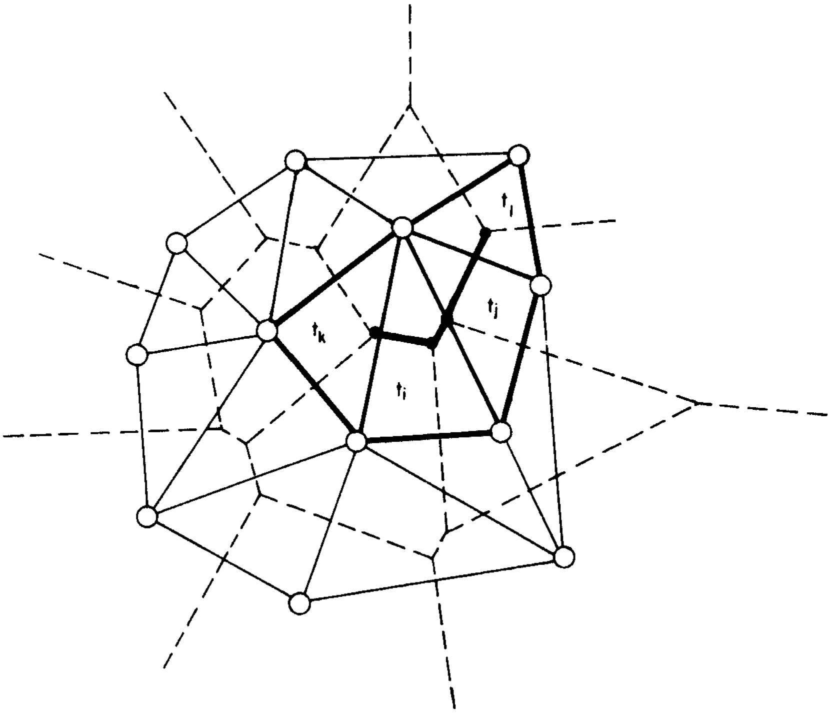


FIG. 5. A nearest neighbor triangle quadruple (t_i, t_j, t_k, t_l).

Although the length of the Voronoi edge assists in defining candidate triangle pairs, additional information is needed concerning whether a particular triangle pair will reduce the overall length of the $\widehat{\mathrm{SMT}}$ of V. The following definitions and lemma are needed:

Frond. This is a subgraph of the MST of V interconnecting four adjacent vertices of V with exactly three edges of the MST of V. There are, in general, three types of fronds: a sac frond, a star frond, and a zee frond. These are illustrated in Figure 6.

Since the frond represents a disjoint minimum spanning tree of V, any reduction in the length of the frond will necessarily reduce the overall length of the $\widehat{\mathrm{SMT}}$ of V.

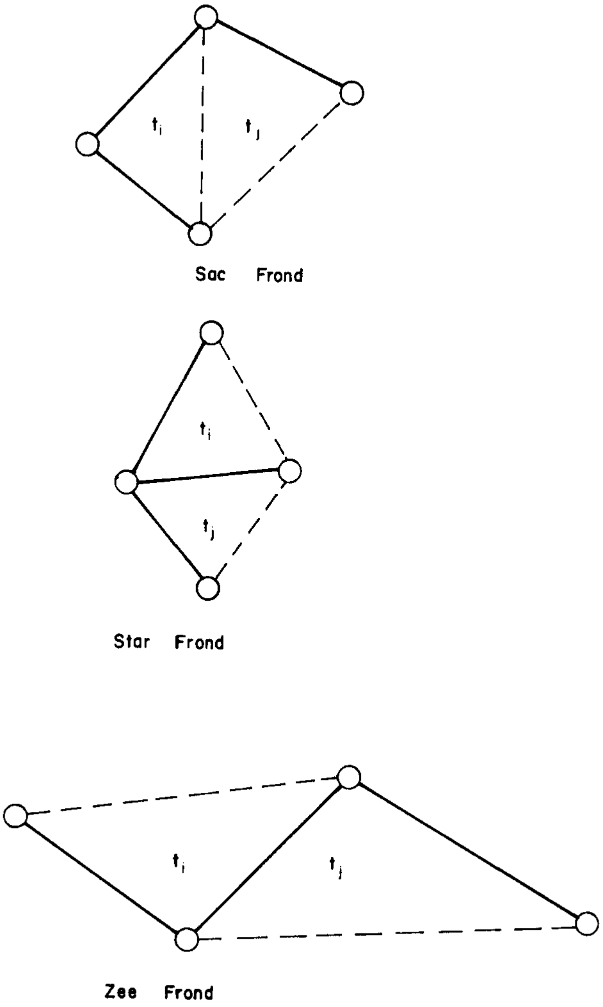


FIG. 6. Alternative frond configurations.

Lemma 8. Given the vertices of two adjacent Delaunay triangles (t_i, t_j) which form a frond, then (t_i, t_j) can be concatenated into the $\mathrm{S\hat{M}T}$ of V if and only if all the vertices of the frond are in disjoint sets of the $\mathrm{S\hat{M}T}$ of V.

Proof: The $\mathrm{S\hat{M}T}$ of V is a spanning tree of V by definition. If at least two of the vertices of (t_i, t_j) are in the same set of the $\mathrm{S\hat{M}T}$ of V, then concatenating the triangles will create an additional link between the vertices in the same set of the $\mathrm{S\hat{M}T}$ of V. This will create a cycle in the $\mathrm{S\hat{M}T}$ of V. Therefore if two or more vertices of (t_i, t_j) are in the same set of $\mathrm{S\hat{M}T}$ of V, (t_i, t_j) cannot be concatenated. In a similar manner, only if vertices of (t_i, t_j) are in disjoint sets can (t_i, t_j) be concatenated, otherwise one would have created a cycle in the $\mathrm{S\hat{M}T}$ of V. Q.E.D.

V. GENERAL CONCATENATION OVERVIEW

Assuming that the Delaunay triangulation for the specific L_2 metric has been created, the concatenation procedure is as follows:

Step 1: Construct the MST on the triangulation. This is actually a subgraph of the Delaunay triangulation.

Step 2: Mark the triangles during the MST construction process and identify those with two edges of the MST. Place those triangles with two edges of the MST in a priority queue. Call this priority queue Q.

Step 3: Where possible, concatenate pairs of triangles, and, if not, add the single triangle with its Steiner point and the three edges into the overall $\mathrm{S\hat{M}T}$ of V.

Step 4: The concatenation process is complete once the priority queue is empty.

A flow chart of the overall concatenation process is illustrated in Figure 7. Once the MST is constructed, the $\mathrm{S\hat{M}T}$ can be found in linear time, since it is necessary to pass through the priority queue of marked triangles only once. In the worst case, all the triangles of the Delaunay triangulation would be in the queue.

One important point that should be mentioned is that not just any triangulation is usable here, because not all triangulations will have the MST as a subgraph of its edge set. There may be triangulations which have the MST as a subgraph of its edge set, other than the Delaunay triangulation. Because the MST is a subgraph of the Delaunay triangulation, we can guarantee a worst case $O(n \log n)$ time step for constructing the MST.

VI. COMPLEXITY ANALYSIS

Since aspects of the worst case behavior of the $\mathrm{ES\hat{M}T}$ algorithm have already been discussed, this analysis will be primarily a brief summary.

Construction of the Delaunay triangulation takes $O(n \log n)$ time in the worst case. Constructing the MST takes $O(n \log n)$ time as defined in Step 2 of the concatenation process, although if implemented differently construction of the MST would take $O(n)$ time. Creation of the list of nearest triangle neighbors is done in linear time; and if implemented properly, the behavior of the disjoint-set-union algorithm is linear in the number of edges of the Delaunay triangulation. Since the number of edges in Delaunay triangulation is $3(n - 1) - C$, the disjoint-set-union procedure is linear in the number of vertices of V. The remainder of the loops of the concatenation proce-

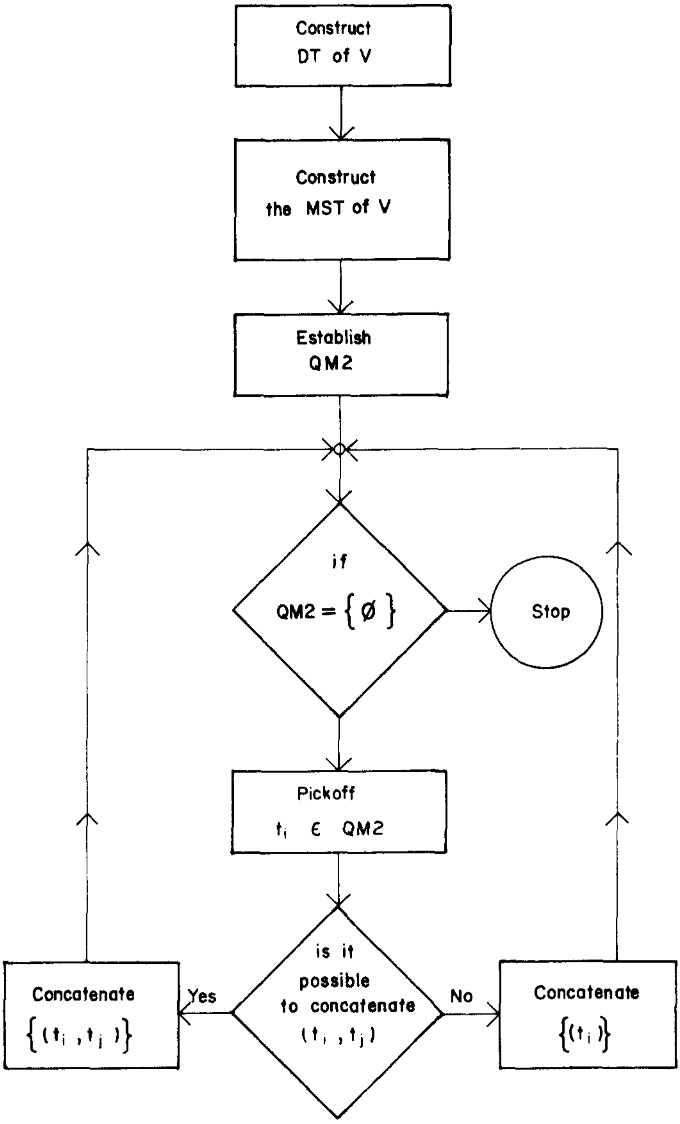


FIG. 7. Flow chart for general concatenation process.

dure require constant time independent of the size of V. Finally, given the MST, creation of the priority queue of marked triangles takes $O(\log t_i)$ time where t_i represents the number of triangles with two edges in the MST. The single pass through the concatenation algorithm therefore takes $O(t_i \log t_i)$ time. In the worst case all triangles of the Delaunay triangulation would be in the queue. Thus, since the number of triangles is $2(n - 1) - C$, the overall concatenation algorithm is $O(n \log n)$.

The total amount of storage required is proportional to the number of edges in the Delaunay triangulation. Since there are exactly $3(n - 1) - C$ edges in the triangulation, the total amount of storage is $O(n)$.

VII. 14 ESMT COMPUTATION RESULTS

In order to evaluate the import of the $O(n \log n)$ alogrithm and its computational performance and solution accuracy, the $O(n \log n)$ version was coded in FORTRAN IV and tested on a number of uniformly distributed sets of points in E^2 bounded by a unit square. The same random points generated for the test problems of the previous $O(n^4)$ algorithm were utilized to test the $O(n \log n)$ algorithm. As we shall see in Figure 8, there is a remarkable change in the running time of the new algorithm over the previous algorithm. Before examining the running times of the new algorithm, however, we first ought to make sure that the percentage $\widehat{\mathrm{SMT}}$/MST reductions of the $O(n \log n)$ algorithm are at least as good as the $O(n^4)$ algorithm. A simple t-test for the $n = 10$ point problems was utilized. Table I illustrates the percentage reduction ($\widehat{\mathrm{SMT}}$/MST) for the two different algorithms as well as their differences.

Calculation of the t statistic, t_0, shows that no significant difference existed at the 90% confidence level with fourteen degrees of freedom. Even though the overall percentage reduction for the $O(n \log n)$ algorithm is 3.137% as compared to a value of 2.874% for the $O(n^4)$ algorithm, no significant difference occurs statistically. Therefore, all we can say at this point is that the new algorithm is at least as good as the old one in terms of its solution efficacy. Nonetheless, this is encouraging because one should not necessarily expect that the faster algorithm would achieve greater reductions. The $O(n^4)$ algorithm considers more Steiner points and different possible $\widehat{\mathrm{SMT}}$ configurations.

Table II summarizes the overall performance of the $O(n \log n)$ algorithm on point set sizes ranging from 10 to 50 points.

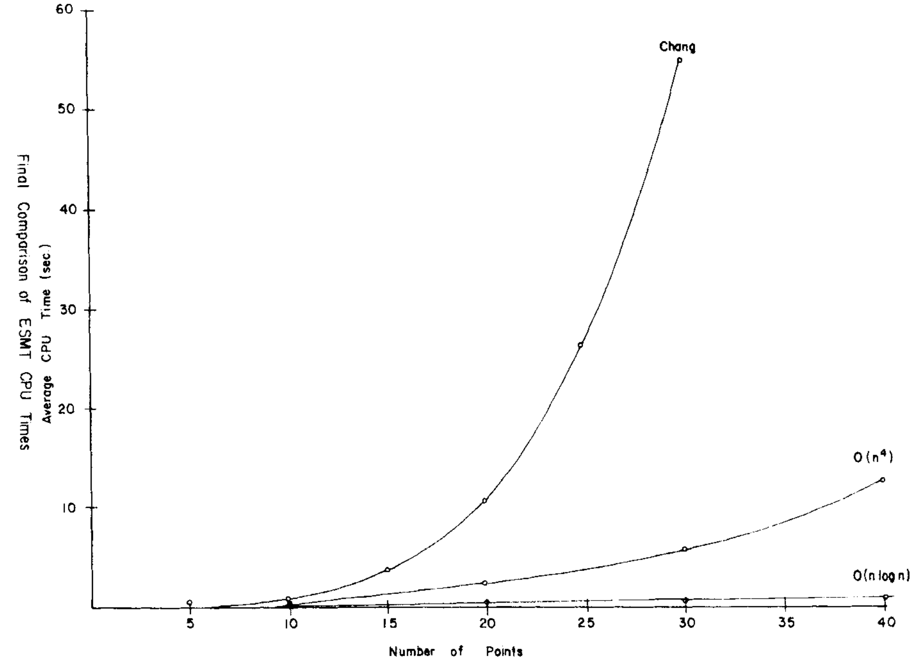


FIG. 8. Final comparison of ESMT CPU times.

TABLE I. ESMT algorithm $\hat{SMT}$/MST percentage reductions.

Observed results	$O(n^4)$	$O(n \log n)$	Algorithm difference
Data set number			
1	3.88	4.38	0.50
2	3.27	3.65	0.38
3	2.61	2.61	0.00
4	1.22	1.22	0.00
5	2.81	2.81	0.00
6	1.17	1.81	0.64
7	2.88	2.88	0.00
8	4.77	6.85	2.08
9	7.30	7.23	-0.07
10	2.25	0.91	-1.34
11	4.70	5.37	0.67
12	0.90	1.39	0.49
13	3.97	3.97	0.00
14	1.05	1.05	0.00
15	0.33	0.93	0.60
Mean	2.87	3.14	0.27
Variance of differences $s^2 = 0.532$			
t-test statistic, $t_0 = 0.729$, not significant at 90% level.			

TABLE II. Experimental results for the ESMT algorithm.[a]

Number of nodes	CPU time in seconds (SD)	$\hat{SMT}$/MST mean percent decrease (SD)	Maximum percent decrease	Mean number of Steiner points (SD)	Maximum number of Steiner points
10	0.293 (.015)	3.173 (2.09)	6.847	3 (1)	4
20	0.569 (.015)	2.333 (0.70)	4.227	5.6 (0.8)	7
30	0.799 (.022)	2.769 (0.89)	4.554	9.1 (1.7)	12
40	1.089 (.032)	2.663 (0.64)	4.014	13.2 (1.8)	16
50	1.375 (.025)	2.568 (0.57)	3.443	16 (1.8)	19

[a] Fifteen runs of each size problem were run on a DEC-10 computer housed at the Coordinated Science Laboratory at the University of Illinois, Urbana campus.

Certainly, the results of this paper would be strengthened if we could indicate a bound for our algorithm on how close it will come to the optimal Steiner tree for the given point set. However, there is no known bound on the ratio of the lengths of heuristic Steiner minimal trees and optimal Steiner trees, for either the L_1 or L_2 metrics [10]. This remains a question for further research.

VIII. SUMMARY AND CONCLUSIONS

We have presented in this paper an $O(n \log n)$ algorithm for generating heuristic solutions to the Steiner Minimal Tree (SMT) problem on the Euclidean metric. The overall strategy used was a decomposition approach where we utilized the Delaunay triangulation to "decompose" the point set and we used the properties of the Voronoi diagram and the Minimum Spanning Tree (MST) of the point set to "recompose" the solution for the point set. Along with a detailed presentation of the algorithm, computational running times were compared with a previous $O(n^4)$ algorithm. The results not only indicate a significant reduction in running time but also that the solutions with the $O(n \log n)$ algorithm are at least as good as the $O(n^4)$ algorithm in reducing the length of the $\hat{SMT}$ over the MST. Future research will explore the development of better bounds for how close to the optimal solution this heuristic procedure will come. In general, the approach utilizing the Delaunay triangulation and the properties of the Voronoi diagram opens up new vistas on developing algorithms for generalized $\hat{SMT}$ problems.

The authors would like to thank the referees for their valuable criticism and the Graduate Research Board of the University of Illinois for providing the necessary funds for conducting the computer experiments.

References

[1] A. H. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] S. K. Chang, "The Generation of Minimal Trees with a Steiner Topology," *JACM*, **19**, 699-711 (1972).

[3] F. R. K. Chung and F. K. Hwang, "A Lower Bound for the Steiner Tree Problem," *SIAM J. Appl. Math.*, **34**, 27-36 (1978).

[4] D. R. Cheriton and R. E. Tarjan, "Finding Minimum Spanning Trees," *SIAM J. Comput.*, **5**, 724-742 (1976).

[5] D. R. Courant and H. Robbins, *What Is Mathematics?* Oxford University Press, New York, 1941.

[6] R. Drysdale and D. T. Lee, "Generalized Voronoi Diagram in the Plane," *Proc. 16th Allerton Conference on Comm., Control and Computing* (1978) pp. 833-842.

[7] M. R. Garey, R. L. Graham, and D. S. Johnson, "The Complexity of Computing Steiner Minimal Trees," *SIAM J. Appl. Math.*, **32**, 835-859 (1977).

[8] E. N. Gilbert and H. O. Polak, "Steiner Minimal Trees," *SIAM J. Appl. Math.*, **16**, 1-29 (1968).

[9] R. L. Graham and F. K. Hwang, "Remarks on Steiner Minimal Trees I," *Bull. Inst. Math. Acad. Sinica*, **4**, 177-182 (1976).

[10] F. K. Hwang, "An $O(n \log n)$ Algorithm for Suboptimal Rectilinear Steiner Trees," *IEEE Circuits and Systems*, **CAS-26**, 75-77 (1979).

[11] F. K. Hwang, "The Rectilinear Steiner Problem," *J. Design Auto. Fault Toler. Anal.*, **2**, 303-310 (1978).

[12] R. J. Karp, "Probabilistic Analysis of Partitioning Algorithms for the Traveling Salesman Problem in the Plane," *Math. Oper. Res.*, **2**, 209-244 (1977).

[13] H. W. Kuhn, "A Note on Fermat's Problem," *Math. Program.*, **4**, 90-107 (1973).

[14] H. W. Kuhn, "Steiner's Problem Revisited," in *Studies in Mathematics, Vol. 10*, Studies in Optimization, G. B. Dantzig and B. C. Eaves, Eds., The Math. Assoc. Am., 1975.

[15] D. T. Lee, "On Finding *k* Nearest Neighbors in the Plane," Technical Report, Department of Computer Science, University of Illinois, 1976.

[16] D. T. Lee, "Generalization of Voronoi Diagrams," Extended abstract, Coordinated Science Laboratory, University of Illinois (unpublished).

[17] Z. A. Melzak, "On the Problem of Steiner," *Canad. Math. Bull.*, **4**, 143-148 (1961).

[18] Z. A. Melzak, *Companion to Concrete Mathematics*, Wiley, New York, 1973.
[19] Z. A. Melzak, *Mathematical Ideas, Modeling, and Applications*, Wiley, New York, 1976.
[20] W. Miehle, "Link-Length Minimization in Networks," *Oper. Res.*, 6, 232-243 (1958).
[21] H. O. Polak, "Some Remarks on the Steiner Problem," *J. Combinatorial Theory*, A-24, 278-295 (1978).
[22] M. I. Shamos, "Geometric Complexity," *Seventh Annual ACM SIGACT Conference*, 1975, pp. 224-233.
[23] M. I. Shamos, "Computational Geometry," Ph.D. thesis, Yale University, 1977.
[24] J. MacGregor Smith and J. S. Liebman, "Steiner Trees, Steiner Circuits, and the Interference Problem in Building Design," *Eng. Optim.*, 4, 15-36 (1979).
[25] J. MacGregor Smith, "Algorithms for Generalized Steiner Network Problems," Ph.D. thesis, Department of Mechanical and Industrial Engineering, University of Illinois, 1978 (unpublished).

Received December 1978
Accepted June 1980