An $O(n \log n)$ Algorithm for Rectilinear Minimal Spanning Trees

F. K. HWANG

Bell Laboratories, Murray Hill, New Jersey

ABSTRACT. Let P be a set of points in the plane with rectilinear distance An $O(n \log n)$ algorithm for the construction of a Voronoi diagram for P is given By using previously known results, a minimal spanning tree for P can be derived from a Voronoi diagram for P in linear time.

KEY WORDS AND PHRASES minimal spanning tree, rectilinear distance, Voronoi diagram

CR CATEGORIES: 5 32

1. *Introduction*

A minimal spanning tree for a set of n points P in the plane is a minimum length interconnecting tree which has P as its vertex-set. The problem of finding a minimal spanning tree for P has been solved for arbitrary distances between the vertices by Boruvka [2], Choquet [4], Kruskal [5], and Prim [6], and an $O(n^2)$ algorithm [1] is known. Recently, Shamos and Hoey [7] have made use of some special properties of Euclidean distance to develop an $O(n \log n)$ algorithm for the construction of a Voronoi diagram for P and then derive a minimal spanning tree from the Voronoi diagram in less than $O(n \log n)$ time. While deriving a minimal spanning tree from a Voronoi diagram is valid for arbitrary distances, the algorithm for Voronoi diagrams depends critically on the distance function. In this paper we give an $O(n \log n)$ algorithm for Voronoi diagrams when the distance is the rectilinear distance—and hence, an $O(n \log n)$ algorithm for rectilinear minimal spanning trees.

2. *Some Preliminary Remarks*

Consider a set P of points in the plane where the distance is *rectilinear* distance. Namely, if a point is identified by its Cartesian coordinates, i.e. $p_i = (x_i, y_i)$, then the length of the edge $[p_i, p_j]$ is $|x_i - x_j| + |y_i - y_j|$. A bisector of p_i and p_j, denoted by $B(p_i, p_j)$, partitions all the points into two regions $h(i, j)$ and $h(j, i)$ such that $h(i, j)$ contains all the points closer to p_i and $h(j, i)$ contains all the points closer to p_j. When $|y_i - y_j| \neq |x_i - x_j|$, the bisector is unique and assumes one of the four standard forms as shown in Figure 1. When $|y_i - y_j| = |x_i - x_j|$, the bisector is not unique. To be definite, we choose the standard form shown in Figure 1(c) or (d) as a representative.

Note that bisectors consist of only horizontal, vertical, and 45° line segments. Furthermore, a bisector consists of at most three line segments.

Bisectors in Figure 1(a) and (b) are called *horizontal* and those in (c) and (d) *vertical*. In the degenerate case $x_i = x_j$ (or $y_i = y_j$), a horizontal (or vertical) bisector becomes a horizontal (vertical) line.

Author's address Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

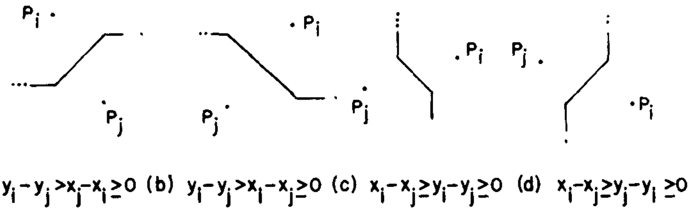

FIG. 1 (a) $y_i - y_j > x_j - x_i \geq 0$, (b) $y_i - y_j > x_i - x_j \geq 0$, (c) $x_i - x_j \geq y_i - y_j \geq 0$, (d) $x_i - x_j \geq y_j - y_i \geq 0$

$VD_i(P) = \bigcap_j h(i, j)$ is defined to be the polygon (with bisectors or their portions as its sides) that contains all the points closer to p_i than to any other $p_j \in P$ and some points (determined by which representative bisector is chosen) as close to p_i as to any other p_j. The *Voronoi* diagram for P, VD(P), is the partition of the plane into n such polygons, some of which are unbounded. Figure 2 shows a Voronoi diagram for seven points.

We consider two linear orderings of points in the plane. Let $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$. Then p_i is said to be *larger* than p_j, denoted by $p_i > p_j$, if either

$$x_i + y_i > x_j + y_j$$

or

$$x_i + y_i = x_j + y_j,$$

and

$$x_i > x_j.$$

Let us define $y_i - x_i$ to be the *M-value* of the point p_i. Then p_i is said to *majorize* p_j, denoted by $p_i \succ p_j$, if the M-value of p_i is greater than the M-value of p_j. If the M-values of p_i and p_j are equal, then we write $p_i \overset{M}{=} p_j$. If the M-value of p_i is greater than or equal to that of p_j, then we write $p_i \succeq p_j$.

A continuous line in the plane is said to be *M-value-monotone* if the M-value from one end to the other is monotone increasing or monotone decreasing. The end with the smaller M-value will be called the *minor end* and the other the *major end*. We denote the minor end of a line l by *Minor* l. From Figure 1 it is easy to verify that a bisector is an M-value-monotone line. Furthermore, if we orient a bisector in its M-value-increasing direction, then the larger of the two points always stays on the right-hand side of the bisector. Let

$$\text{Minor } B(p_3, p_4) \rightarrow B(p_1|p_2)$$

mean that either the minor end of the bisector $B(p_3, p_4)$ lies on the p_1 side of $B(p_1, p_2)$ or Minor $B(p_3, p_4)$ = Minor $B(p_1, p_2)$. We prove some results concerning the location of the minor end of a bisector.

LEMMA 1. *Let p_1 and p_2 be two points in the plane satisfying the condition that either $p_1 \succ p_2$ or $p_1 \overset{M}{=} p_2$ but $p_2 > p_1$. Let p be an arbitrary point in the plane; then*

(*a*) *Minor $B(p, p_2) \rightarrow B(p_2|p_1)$ except in the case $p \succ p_1$ and either $p_2 > p_1 > p$ or $p > p_1 > p_2$;*

(*b*) *Minor $B(p, p_1) \rightarrow B(p_1|p_2)$ if either $p \succeq p_1$ and $p_2 > p_1 > p$, or $p \succ p_1$ and $p > p_1 > p_2$.*

PROOF. If $B(p_1, p_2)$ is a horizontal bisector, then p_1 occupies the upper half and p_2 the lower half. If $B(p_1, p_2)$ is a vertical bisector, then p_1 occupies the left half and p_2 the right half. Now consider another bisector $B(p_3, p_4)$. Since the minor end of a horizontal bisector always lies in the lower half and the minor end of a vertical bisector always lies in the right half, then usually Minor $B(p_3, p_4) \rightarrow B(p_2|p_1)$ except in certain special cases. It is straightforward to verify that (b) and the exception in (a) belong to these special cases. □

A partition of P into two disjoint subsets P_L and P_S is called an *ordered partition* if every point in P_L is larger than every point in P_S. Consider the partition on VD(P) induced by

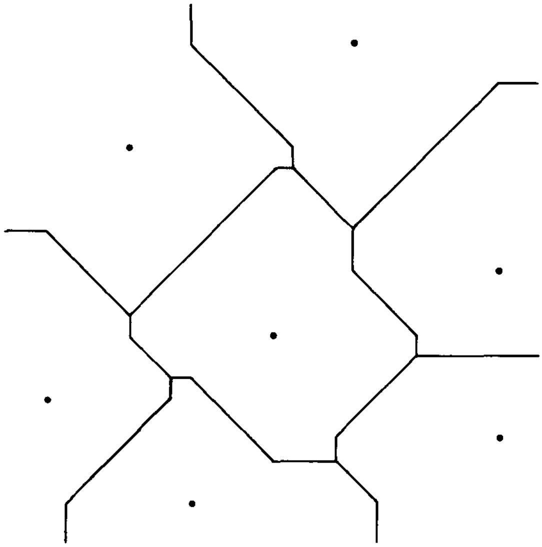

FIG 2

the ordered partition of P into P_L and P_S. Let B be the boundary line that separates $\{VD_i(P) : p_i \in P_L\}$ from $\{VD_i(P) : p_i \in P_S\}$. Then clearly, B is a continuous line consisting of line segments from bisectors $B(p_u, p_v)$ with $p_u \in P_L$ and $p_v \in P_S$. Furthermore, if we orient B in its increasing direction, then all points of P_L lie on its right-hand side. From the facts that bisectors are M-value-monotone and that the larger point always stays on the right-hand side of an increasing bisector, we obtain the following:

LEMMA 2. *B is M-value-monotone.*

Let $\pi_i(L)(\pi_i(S))$ denote the ordered partition of P into P_L and P_S such that p_i is the largest (smallest) point in P_S (P_L). Then the sides of $VD_i(P)$ can be decomposed into two disjoint chains where one chain is a section of the boundary line in the partition $\pi_i(L)$ and the other a section of the boundary line in the partition $\pi_i(S)$. (In case p_i is the largest or smallest point in P, then one of the chains is an empty chain.) Since the boundary lines are M-value-monotone by Lemma 2, we obtain:

LEMMA 3. *The sides of a $VD_i(P)$ can be decomposed into two disjoint chains each of which is M-value-monotone.*

3. *An O(n log n) Algorithm for VD(P)*

We now give a recursive method for constructing the Voronoi diagram V for n given points by first constructing the Voronoi diagrams V_L for the largest $n/2$ points P_L and V_S for the smallest $n/2$ points P_S. V_L and V_S can then be merged into V by constructing B and discarding all lines of V_L below or to the left of B and all lines of V_S above or to the right of B. If B can be constructed in linear time, then this recursive method constructs V in $O(n \log n)$ time. The main content of the section is to show that this is indeed the case.

Our first task in constructing B is to identify two points $p_u \in P_L$ and $p_v \in P_S$ such that Minor $B(p_u, p_v)$ can serve as the minor end of B.

Let m be the set of points in P with the smallest M-value. If

$$P_L \cap m \neq \phi,$$

let p_v be the point with the smallest M-value in P_S (if there are several such points, pick the largest one) and let p_u be the smallest point in P_L among all points in P_L not majorizing p_v. If

$$P_L \cap m = \phi,$$

then let p_u be the point with the smallest M-value in P_L (if there are several, pick the largest one) and let p_v be the largest point in P_S among all points in P_S majorized by p_u. We now show that this choice of p_u and p_v is correct; i.e. Minor $B(p_u, p_v)$ can lie neither within a $\mathrm{VD}_i(P_L)$ for some $p_i \in P_L$ with $p_i \neq p_u$ nor within a $\mathrm{VD}_i(P_S)$ for some $p_i \in P_S$ with $p_i \neq P_v$. Our result is established by an immediate application of Lemma 1. Therefore, our proof consists of the proper selection of a bijection from (p_i, p_u, p_v) to (p, p_1, p_2), which satisfies the conditions of Lemma 1 in each case.

We first consider the case

$$P_L \cap m \neq \phi.$$

Then $p_v \succeq p_u$. This case consists of three subcases:

(i) $p_i \in P_L$ and $p_i > p_v$. Then necessarily, $p_i > p_u$. Set $p = p_v, p_1 = p_i$, and $p_2 = p_u$. Lemma 1(a) applies since $p_i > p_v$.

(ii) $p_i \in P_L$ and $p_v \succeq p_i$. Then $p_i > p_u$ by the definition of p_u. Set $p = p_v$. If $p_i > p_u$, set $p_1 = p_i$ and $p_2 = p_u$. Since $p_i > p_u > p_v$, Lemma 1(a) applies. If $p_i \preceq p_u$, set $p_1 = p_u$ and $p_2 = p_i$. Lemma 1(b) applies since $p_v \succeq p_u$ and $p_i > p_u > p_v$.

(iii) $p_i \in P_L$. Then either $p_i > p_v$, or $p_i \overset{M}{=} p_v$ and $p_i < p_v$. Set $p = p_u, p_1 = p_i$, and $p_2 = p_v$. Lemma 1(a) applies since $p_i \succeq p_u$.

Next we consider the case

$$m \cap P_L = \phi.$$

Then $p_v < p_u$. Again we consider three subcases:

(i) $p_i \in P_L$. Then either $p_i > p_u$, or $p_i \overset{M}{=} p_u$ and $p_i < p_u$. Set $p = p_v, p_1 = p_i$, and $p_2 = p_u$. Since $p_i > p_v$, Lemma 1(a) applies.

(ii) $p_i \in P_S$ and $p_i \overset{M}{=} p_u$. Note that if $p_i \overset{M}{=} p_u \overset{M}{=} p_v$, then $p_i < p_v$ by the definition of p_v. Set $p = p_u, p_1 = p_i$, and $p_2 = p_v$. Lemma 1(a) applies since $p_i \succeq p_u$.

(iii) $p_i \in P_S$ and $p_i < p_u$. Then $p_i < p_v$. Set $p = p_u$. If $p_i = p_v$, set $p_1 = p_i$ and $p_2 = p_v$. Since $p_u > p_v > p_i$, Lemma 1(a) applies. If $p_i < p_v$, set $p_1 = p_v$ and $p_2 = p_i$. Since $p_u > p_v$ and $p_u > p_v > p_i$, Lemma 1(b) applies. The proof is completed.

Clearly, there exists a linear procedure to find such p_u and p_v. For example, we first construct three nonempty sets m, m_L, and m_S such that m (m_L, m_S) consists of all the points in P (P_L, P_S) that have the smallest M-value among all points in P (P_L, P_S). Each such set can be constructed in linear time since it takes linear time to find one point in P (P_L, P_S) with the smallest M-value; then all points in P (P_L, P_S) with the same M-value can be picked up in one pass. We also construct a set $m_{S'}$ which consists of all points in P_S with M-values less than the M-value of the points in m_L in one pass of P_S. Note that $m_{S'}$ is nonempty if $m = m_S$. Let w and w' be the largest and the smallest point in m_L, respectively. Furthermore, let z be the largest point in m_S and z' the largest point in $m_{S'}$. Then w, w', z, and z' can be found in linear time. Now if m does include a point from V_L, set $v = z$ and $u = w'$; if not, set $v = z'$ and $u = w$.

After we set Minor B = Minor $B(p_u, p_v)$, we extend B in the intersection of $\mathrm{VD}_u(P_L)$ and $\mathrm{VD}_v(P_S)$ until it intersects a side of either Voronoi polygon. If B first intersects a line segment of $\mathrm{VD}_u(P_L)$, which is a part of $B(p_u, p_{u'})$ for some $p_{u'} \in P_L$, then B enters the intersection of $\mathrm{VD}_{u'}(P_L)$ and $\mathrm{VD}_v(P_S)$ and continues with $B(p_{u'}, p_v)$. Similarly, if B first intersects a line segment of $\mathrm{VD}_u(P_L)$, which is a part of $B(p_v, p_z)$ for some $p_z \in P_S$, then B enters the intersection of $\mathrm{VD}_u(P_L)$ and $\mathrm{VD}_z(P_S)$ and continues with $B(p_u, p_z)$ In the rare case that B intersects a line segment from $B(p_u, p_{u'})$ and a line segment from $B(p_v, p_z)$ simultaneously, then B enters the intersection of $\mathrm{VD}_{u'}(P_L)$ and $\mathrm{VD}_z(P_S)$ and continues with $B(p_{u'}, p_z)$. However, we can always treat the last case as a degenerate case of the two former cases with a zero length bisector.

Therefore, our next task is to find which line segment the current extension of B, call it $e(B)$ ($e(B)$ is a bisector growing from its minor end toward its major end), is going to intersect first Suppose $e(B)$ lies in the intersection of $\mathrm{VD}_i(P_L)$ and $\mathrm{VD}_j(P_S)$ From Lemma

3, the sides of $VD_i(P_L)$ and $VD_j(P_S)$ can be decomposed into (at most) four M-value-monotone chains. Let C_1, C_2, C_3, C_4 be these four chains where $C_i = \{C_{i1} = C_{i2} = \cdots\}$ for $i = 1, 2, 3, 4$ and C_{ij}'s are the ordered line segments in the ith chain. Let m_{ij} denote the M-value of the minor end of C_{ij}. Then clearly m_{ij} is monotone increasing in j. We inspect C_{ij} in the increasing order of m_{ij} until an intersection with $e(B)$ is identified. Namely, we inspect the C_{ij} with the smallest m_{ij} first (if there are several such C_{ij}'s pick the one that has the smallest M-value for its major end), and then the second smallest, and so on. Note that when we are inspecting C_{ij}, we know the first intersection point of $e(B)$ (the one with the smallest M-value) must have an M-value not less than m_{ij}. Therefore, all line segments that have M-values less than or equal to m_{ij} for its major end can be discarded since B is M-value-monotone increasing. In particular, we can discard $C_{i(j-1)}$, if it is not already discarded, when C_{ij} is inspected. To facilitate accounting, we will assume that $C_{i(j-1)}$ is discarded at the moment C_{ij} is being inspected.

When an intersection of $e(B)$ with some line segment is identified, we go on inspecting other C_{ij}'s whose m_{ij} does not exceed the M-value of the identified intersection point. It should be noted that there are at most three such C_{ij}'s (one from each of the other three chains). Then we select the intersection point with the smallest M-value to be the turning point of B; i.e. B takes on a new bisector at this point. Therefore, we have a new $e(B)$ and also a new set of four chains (some possibly overlapping with the old chains) to be inspected. A similar process goes on to completion.

Note that except for (at most) the first four line segments (one from each chain), every time we inspect a line segment C_{ij}, we are able to discard one other line segment, i.e. $C_{i(j-1)}$. When an intersection is identified, then an additional line segment $e(B)$ can be discarded. Let b be the number of line segments in V_L, V_S, and B. Then B can be constructed in $O(b)$ time.

Let $\bar{V}$ be the graph defined by the following operation on V: The vertex-set of $\bar{V}$ is P and there is an edge between vertices P_i and P_j in $\bar{V}$ if and only if $B(P_i, P_j)$ is in V. Then $\bar{V}$ is a planar graph on n vertices and so has no more than $3n - 6$ edges (see [7]). Since the edges of $\bar{V}$ and bisectors of B are in a one-to-one correspondence, V has at most $3n - 6$ bisectors, or at most $9n - 18$ line segments. Therefore, B can be constructed in linear time.

4. *Deriving A Rectilinear Minimal Spanning Tree from VD(P)*

Consider the graph $\bar{V}$ as defined in Section 3. Suppose the edge $[p_i, p_j]$ is not in $\bar{V}$. Then there must exist a point $p_k \in P$ such that the length of the edge $[p_i, p_j]$ is greater than that of either edge $[p_j, p_k]$ or $[p_k, p_j]$ Therefore, the edge $[p_i, p_j]$ will not be used in a rectilinear minimal spanning tree. This proves that a rectilinear minimal spanning tree on $\bar{V}$ is a rectilinear minimal spanning tree for P.

There exist a number of minimal spanning tree algorithms (for graphs) that require no more than $O(n \log n)$ time when the given graph contains no more than $O(n)$ edges. For example, Cheriton and Tarjan [3] give an algorithm that requires linear time for planar graphs. Since $\bar{V}$ has only $O(n)$ edges, a rectilinear minimal spanning tree on $\bar{V}$ can be constructed in $O(n \log n)$ time.

Note that a Voronoi diagram of a rectilinear minimal spanning tree on P implies a sorting of P when P is projected onto a horizontal line. Since $O(n \log n)$ is a lower bound on sorting algorithms, it is also a lower bound on constructing Voronoi diagrams and rectilinear minimal spanning trees.

REFERENCES

1 AHO, A V, HOPCROFT, J E, AND ULLMAN, J D *The Design and Analysis of Computer Algorithms* Addison-Wesley, Reading, Mass, 1974, p 470

2 BORUVKA, O On a minimal problem *Prace Moraské Pridovedecké Spolecnosti, Vol 3*, 1926

3 CHERITON, R, AND TARJAN, R E Finding minimum spanning trees *SIAM J Compting 5* (Dec 1976), 724–742

4 CHOQUET, G Etude de certains réseaux de routes *C R Acad Sci Paris 206* (1938), 310–313

5 KRUSKAL, J B On the shortest spannıng subtree of a graph. *Proc. Amer Math. Soc* 7 (Feb 1956), 48–50
6 PRIM, R C Shortest connectıng networks and some generalızatıons *Bell Syst Tech J* *36* (Nov 1957), 1389–1401
7 SHAMOS, M. I., AND HOEY, D Closest poınt problems Proc 16th Annual Symp Foundatıons of Comptr Scı., 1975, pp 151–162 (avaılable from IEEE, New York)

RECEIVED MARCH 1977, REVISED JULY 1978