

## There Are Planar Graphs Almost as Good as the Complete Graph

L. PAUL CHEW\*

*Department of Math and Computer Science, Dartmouth College,  
Hanover, New Hampshire 03755*

Received July 15, 1987; revised November 20, 1987

Given a set  $S$  of points in the plane, the complete graph on  $S$  (the graph with an edge connecting each pair of points) is the only graph with the property that for any points  $A$  and  $B$  of  $S$  there exists an  $A$ -to- $B$  path along edges of the graph with path length equal to the straight-line distance between  $A$  and  $B$ . We show there is a *planar* graph  $G$  on  $S$  with a similar property: for any points  $A$  and  $B$  of  $S$ , there exists an  $A$ -to- $B$  path along edges of  $G$  with path length at most  $2|AB|$ , where  $|AB|$  is the Euclidean straight-line distance between  $A$  and  $B$ . If  $S$  is of size  $n$  then, because  $G$  is planar,  $G$  has just  $O(n)$  edges instead of the  $O(n^2)$  edges required for the complete graph. The graph  $G$  that has this property is a type of *Delaunay triangulation*. Applications include network design and motion planning. © 1989 Academic Press, Inc.

### 1. INTRODUCTION AND BACKGROUND

Let  $S$  be a set of  $n$  points in the plane. One way to design a network on  $S$  in which transmission distances are small is to use the complete graph, the graph with an edge connecting each pair of points in  $S$ . The advantage of using this complete network is that the transmission distance between any two points of  $S$  is as small as possible.

In this paper, we show there is a planar network on  $S$  in which transmission distances are at most 2 times the optimal distance (the distance in the complete network). Because the network is planar it has  $O(n)$  edges instead of the  $O(n^2)$  edges needed for the complete network. The planar network that has this property is a special kind of *Delaunay triangulation* of the set  $S$ .

A *Delaunay triangulation* of a set  $S$  of points in the plane is most easily introduced by reference to the *Voronoi diagram* of  $S$ . (See Fig. 1.) The Voronoi diagram for  $S$  divides the plane into regions, one region for each point in  $S$ , such that for each region  $R$  and its corresponding point  $p$ , every point within  $R$  is closer to  $p$  than to any other point of  $S$ . The boundaries of these regions form a planar graph. The Delaunay triangulation of  $S$  is the straight-line dual of the Voronoi diagram for  $S$ ; that is, we connect a pair of points in  $S$  if they share a Voronoi boundary. The Voronoi diagram and its dual, the Delaunay triangulation, have been found to

\* Current address: Computer Science Department, Cornell University, Ithaca, New York 14853.

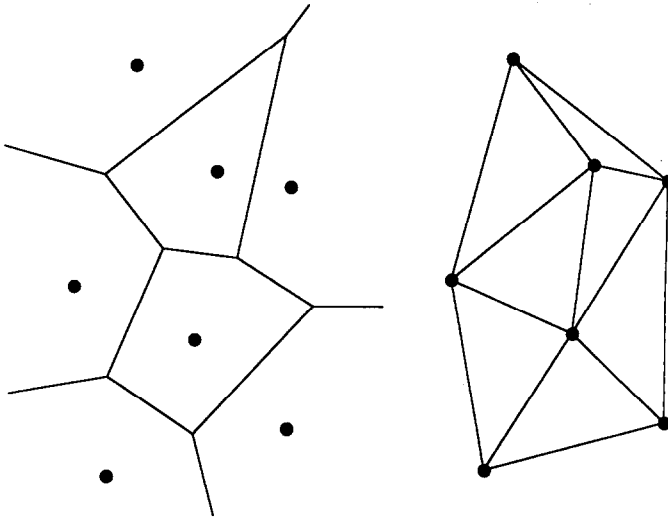


FIG. 1. A Voronoi diagram and the corresponding Delaunay triangulation.

be among the most useful data structures in computational geometry. (See [PS85] for a large number of Voronoi diagram applications.)

The results presented here use a Delaunay triangulation based on a different distance function than the standard Euclidean distance. We use a *convex distance function* based on an equilateral triangle. Convex distance functions, also called *Minkowski distance functions*, were first used by Minkowski in 1911. For such a function, distance can be defined in terms of a unit *circle*. *Circle*, here, is printed in italics because this circle can be any convex shape. To find the distance from point  $p$  to point  $q$ , we center the unit *circle* at point  $p$  and expand (or contract) the *circle* until its boundary intersects  $q$ . By definition, the distance from  $p$  to  $q$  is the factor by which the *circle* changed. If the *circle* is a true circle, with its *center* in the usual place, then we get the usual Euclidean distance. If the *circle* is an arbitrary convex shape with a *center* anywhere in its interior then we get a convex distance function. Note that the distance defined in this manner is not necessarily a metric, since the symmetry property (the distance from  $p$  to  $q$  is the same as the distance from  $q$  to  $p$ ) holds only if the given *circle* is symmetric about its center.

The convex distance function used in this paper has an equilateral triangle, oriented as in Fig. 2, as the distance-defining convex shape. (Since we are concerned only with relative distances, any size equilateral triangle will do.) The TD (equilateral triangle convex distance function) Voronoi diagram is defined just like the standard (Euclidean) Voronoi diagram except the equilateral triangle convex distance function is used to calculate distances. (See Fig. 3.) A useful intuition is to think of circular waves (i.e., waves in the shape of the distance defining *circle*) expanding simultaneously from each point in the set  $S$ ; where the waves collide, we have a Voronoi boundary. Chew and Drysdale show that, like the Euclidean

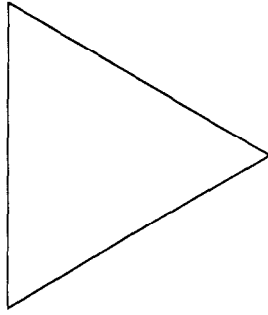


FIG. 2. The distance defining circle.

Voronoi diagram, the boundaries of the TD Voronoi regions form a planar graph and the TD Voronoi diagram (and Voronoi diagrams based on other convex distance functions) can be constructed in  $O(n \log n)$  time where  $n$  is the number of points in the set  $S$ . See [CD85] for more information on convex distance functions and their relationship to Voronoi diagrams and Delaunay triangulations.

The TD Delaunay triangulation can be derived from the corresponding Voronoi diagram in  $O(n)$  time, or, alternately, it can be built directly using a method similar to the Euclidean-case method presented by Lee and Schachter [LS80]. Perhaps the best way to build a TD Delaunay triangulation is to use a method similar to the sweep-line technique developed by Fortune [Fo87] for the standard Voronoi diagram.

An important property of Delaunay triangulations is that for any edge of the triangulation there exists an empty circle through the endpoints of the edge. Indeed, we use this property as part of our definition of Delaunay triangulation.

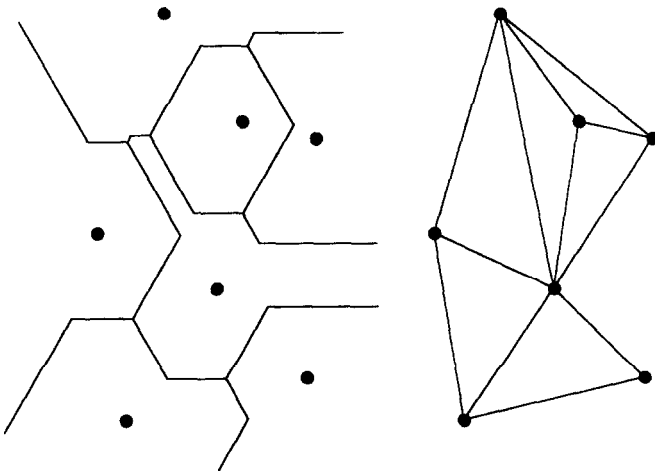


FIG. 3. A TD Voronoi diagram and the corresponding TD Delaunay triangulation.

**DEFINITION.** An edge  $e$  of a straight-line planar graph  $G$  is said to have the *empty circle property* if there exists a circle  $C$  through the endpoints of  $e$  such that  $C$  contains no vertices of  $G$  in its interior. A straight-line planar graph  $G$  is said to have the *empty circle property* if each edge of  $G$  has the empty circle property.

Of course, for the equilateral triangle convex distance function, the circle in question is based on the equilateral triangle described above. An examination of Figs. 3 and 4 should convince the reader that a Delaunay-triangulation empty circle is a reflection of the distance-defining circle.

Basically, a Delaunay triangulation of  $S$  is a triangulation of  $S$  which has the empty circle property. This definition works fine for the standard (Euclidean) Delaunay triangulation, but causes problems for Delaunay triangulations based on convex distance functions, such as the TD Delaunay triangulation. The difficulty is, that under common definitions of *triangulation* (see, for example, [PS85]) the TD Delaunay triangulation is not necessarily a triangulation. For example, the TD Delaunay triangulation in Fig. 4 is not a triangulation because the edge between the two right-most points is missing. Of course, such an edge should not be included in a TD *Delaunay* triangulation because that edge does not have the empty circle property.

To avoid such problems, we define a Delaunay triangulation as a maximal graph with the empty circle property. (By *maximal*, we mean a graph to which no more edges can be added.) This leaves us with some unfortunate terminology (a TD Delaunay triangulation is not necessarily a triangulation), but the properties of a TD Delaunay triangulation are so close to the properties of the standard Delaunay triangulation that it seems worthwhile to retain *Delaunay triangulation* as part of its name.

**DEFINITION.** Let  $S$  be a set of points in the plane. A *Delaunay triangulation* of  $S$  is a maximal straight-line planar graph  $G$  on  $S$  with the empty circle property.

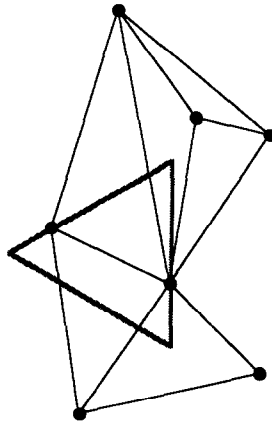


FIG. 4. A TD Delaunay triangulation and an empty circle for one of its edges.

If no 4 points of  $S$  are cocircular (using the appropriate *circle*) then the Delaunay triangulation of  $S$  is unique. For a case in which there is more than one possible Delaunay triangulation then, for our purposes, any of them will do.

We also make use of the following well-known property, often used as the definition of Delaunay triangulation. (See Fig. 5.)

**PROPOSITION.** *Let  $T$  be a Delaunay triangulation of a set  $S$  of points in the plane. For each empty triangle of  $T$  there exists a circle  $C$  such that (1)  $C$  goes through each vertex of the triangle and (2)  $C$  does not contain any points of  $S$  in its interior.*

The following theorem is the main result of this paper. The proof is presented in Section 2.

**THEOREM 1.** *Let  $S$  be a set of points in the plane and let  $T$  be a TD Delaunay triangulation of  $S$ . For any points  $A$  and  $B$  of  $S$ , there exists an  $A$ -to- $B$  path along edges of  $T$  that has length  $\leq 2|AB|$ , where  $|AB|$  is the straight-line Euclidean distance between points  $A$  and  $B$ .*

Theorem 1 is an improvement over earlier work by the author in which the following theorem was proved for the  $L_1$ -metric Delaunay triangulation. (The  $L_1$  metric is a convex distance function in which the unit *circle* is a square tipped at  $45^\circ$ .)

**THEOREM 2 [Ch86].** *Let  $S$  be a set of points in the plane and let  $T$  be an  $L_1$  Delaunay triangulation of  $S$ . For any points  $A$  and  $B$  of  $S$ , there exists an  $A$ -to- $B$  path along edges of  $T$  that has length  $\leq (\sqrt{10})|AB|$ .*

The techniques used in the proof of Theorem 2 are essentially similar to the

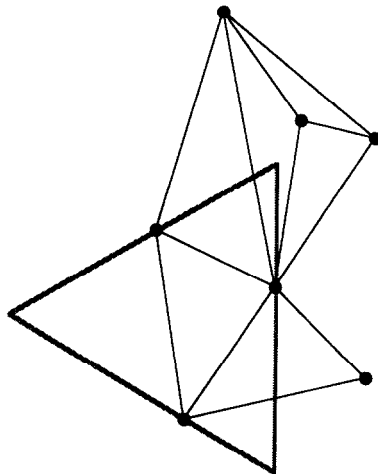


FIG. 5 A TD Delaunay triangulation and an empty circle for one of its triangles.

techniques presented in this paper for the proof of Theorem 1. Readers who are particularly interested in the proof for the  $L_1$  metric are directed to the earlier paper [Ch86].

## 2. THE PROOF

To simplify our presentation, we need to ensure that the Delaunay triangulation covers the entire plane. (See Fig. 6.) To do this we add three points to  $S$ ; these three points correspond to the vertices of a very large version of our distance defining equilateral triangle. Consider the three points  $(w)(2, 0)$ ,  $(w)(-1, \sqrt{3})$ , and  $(w)(-1, -\sqrt{3})$ , and let  $w$  approach  $\infty$ . (Notation: we use  $(w)(x, y)$  to represent the point  $(wx, wy)$ .) In effect, we add three points at *infinity*:  $(\infty)(2, 0)$ ,  $(\infty)(-1, \sqrt{3})$ , and  $(\infty)(-1, -\sqrt{3})$ . By including these points, we avoid a number of special cases in the proofs without affecting the results of the main theorem since points at infinity are never used along finite-length paths.

We start with a lemma about a special case of Theorem 1. The proof of this special case allows us to present the ideas needed for the proof of Theorem 1 without obscuring these ideas with details. Figure 7 shows an example in which Lemma 1 can be applied.

**LEMMA 1.** *Let  $S$  be a set of points in the plane and let  $T$  be the TD Delaunay triangulation of  $S$  (where the orientation of the distance-defining circle is as in Fig. 2). If there are points  $A$  and  $B$  in  $S$  such that segment  $AB$  is horizontal then there exists an  $A$ -to- $B$  path along edges of  $T$  that has length  $\leq (\sqrt{3}) |AB|$ .*

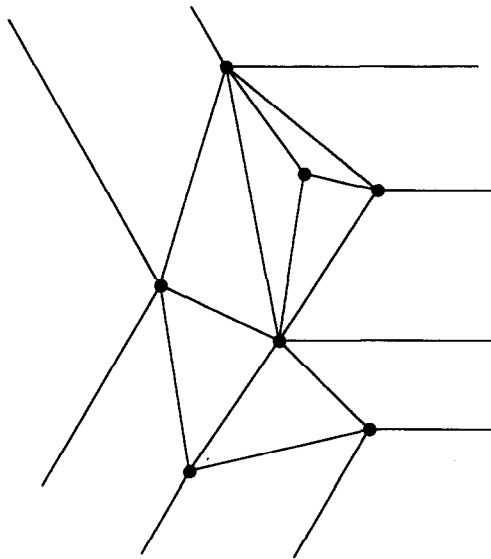


FIG. 6. A TD Delaunay triangulation using points at infinity.

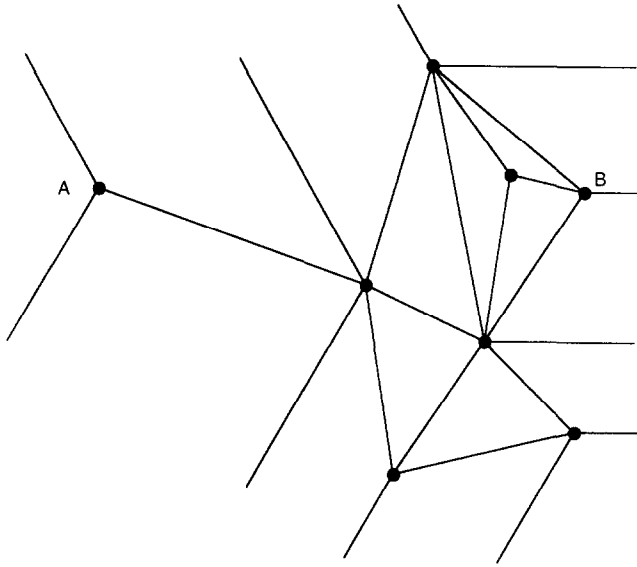


FIG. 7. A TD Delaunay triangulation with  $AB$  horizontal. Points at  $\infty$  have been added.

*Proof.* We prove the existence of the desired path by presenting an algorithm to compute it.

#### ALGORITHM A.

0. Add the three points at infinity (corresponding to the vertices of an equilateral triangle with a leftmost vertical edge) to  $S$  and  $T$ . We consider  $T$  to be a collection of Delaunay triangles; each triangle has a corresponding empty circle (these circles are, of course, equilateral triangles with a rightmost vertical edge). Let  $L$  be the horizontal line segment from  $A$  to  $B$ ; we can switch the names of  $A$  and  $B$  if necessary so that  $A$  is left of  $B$ . Without loss of generality we may assume that no vertices of  $T$  except  $A$  and  $B$  lie on  $L$  (if there were such a vertex, say  $V$ , we could recursively find paths from  $A$  to  $V$  and from  $V$  to  $B$  and put them together to make a path from  $A$  to  $B$ ). Let  $T'$  be the subgraph of  $T$  that includes just the Delaunay triangles of  $T$  that cover  $L$ , as in Fig. 8. Note that  $T'$  consists

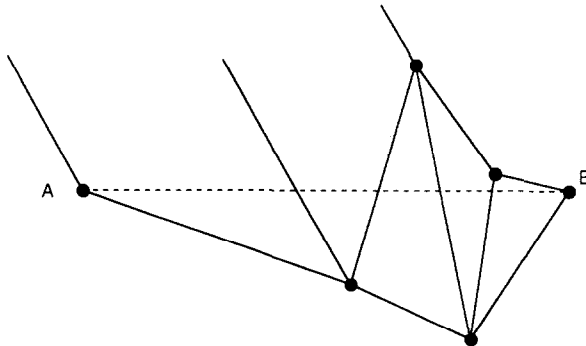


FIG. 8.  $T'$  (i.e., just the triangles that cover segment  $AB$ ).

of a set of triangles ordered from left to right along  $L$ . The remainder of the algorithm uses only vertices and edges of  $T'$ . The desired path is constructed by moving from vertex to vertex of  $T'$ . At each stage we use  $P_i$  to represent our current position (a vertex of  $T'$ ). Initially  $i=0$  and  $P_0$  is the vertex  $A$ .

1. Let  $\Delta$  be the rightmost of all those triangles of  $T'$  that have  $P_i$  as a vertex and let  $C$  be the empty circle (equilateral triangle) through the vertices of  $\Delta$ . Note that by choosing  $\Delta$  to be the rightmost triangle, we ensure that one of the remaining vertices of  $\Delta$  (call it  $X_a$ ) is above  $L$  and clockwise around the boundary of  $C$  from  $P_i$ , and the other vertex of  $\Delta$  (call it  $X_b$ ) is below  $L$  and counterclockwise around the boundary of  $C$  from  $P_i$ . (It is also possible that one of the vertices is actually  $B$  and is thus on  $L$ . We consider the point  $B$  to be both above and below  $L$ .)
2. (See Fig. 9.)
  - if  $P_i$  is on the upper left edge of  $C$
  - then move clockwise around  $C$
  - else if  $P_i$  is on the lower left edge of  $C$
  - then move counterclockwise around  $C$
  - else {  $P_i$  is on the right edge of  $C$  }
  - move toward  $L$ ;
3. Continue in the same direction around  $C$  until either  $X_a$  or  $X_b$  is reached. Whichever is reached call it  $P_{i+1}$ . If  $P_{i+1}$  is  $B$  then Quit else Increment  $i$  and go to step 1.

We use  $P$  to represent the path produced by Algorithm A; its vertices are described by the sequence  $\{P_i\}$  with adjacent vertices connected by edges of  $T'$ . We claim that  $P$  satisfies the requirements of Lemma 1. A series of lemmas is used to prove this claim.

**LEMMA 1.1.** *The triangles used in Algorithm A (called  $\Delta$  in step 1) are ordered from left to right along  $L$ .*

*Proof.* Follows from the use of rightmost triangles in step 1. ■

**LEMMA 1.2.** *Algorithm A terminates, producing a path from  $A$  to  $B$ .*

*Proof.* In step 1,  $X_a$  and  $X_b$  are adjacent to  $P_i$  in  $T'$ , and, in step 3,  $P_{i+1}$  is set to either  $X_a$  or  $X_b$ ; thus, the successive values of  $\{P_i\}$  describe a path within  $T$ .

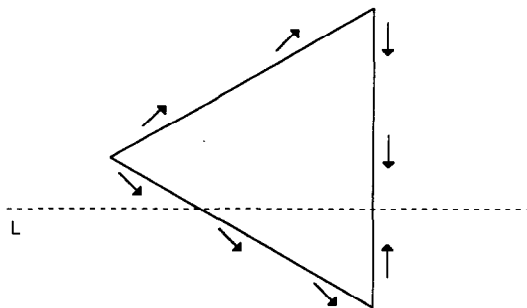


FIG. 9. Directions to move for the different possible positions of  $P_i$  on the circle.



Since there are only finitely many triangles in  $T'$ , it follows from Lemma 1.1 that Algorithm A terminates. ■

The following lemma provides a bound on the length of  $P$ , the  $A$ -to- $B$  path produced by Algorithm A, in terms of a related path  $\Pi$  of a particular shape. The constrained shape of this related path enables us to prove a bound on its length.  $\Pi$  and  $P$  both follow the same sequence  $\{P_i\}$  of vertices, but they use different routes between adjacent vertices. (See Fig. 10.) Note that in steps 1, 2, and 3 of Algorithm A, the next vertex of path  $P$  is chosen by traveling along the boundary of an empty circle.  $\Pi$  is the path made up of these portions of empty circles. Because these circles are equilateral triangles,  $\Pi$  consists of line segments where each line segment is at one of four angles. Taking horizontal as  $0^\circ$ , the possible angles are  $\pm 30^\circ$  and  $\pm 90^\circ$ .

**LEMMA 1.3.** *Let  $P$  be the path along edges of  $T'$  and let  $\Pi$  be the path along the corresponding empty circles.  $|P| \leq |\Pi|$  (i.e., the length of path  $P$  is  $\leq$  the length of path  $\Pi$ ).*

*Proof.* Trivial application of the Triangle Inequality. ■

At this point, we complete the proof of Lemma 1.

*Proof of Lemma 1.* The goal is to prove  $|P|/|L| \leq \sqrt{3}$ . By Lemma 1.3, we are done if we show  $|\Pi|/|L| \leq \sqrt{3}$ . To do this we divide  $\Pi$  and  $L$  into pieces, creating a break wherever  $\Pi$  and  $L$  intersect. We use  $\{\Pi_j\}$  and  $\{L_j\}$  to represent the resulting pieces. Of course, for each value of  $j$ ,  $\Pi_j$  and  $L_j$  share starting and ending points. The desired bound on the ratio is proved by showing that for each  $j$ ,  $|\Pi_j|/|L_j| \leq \sqrt{3}$ . We show this bound holds by studying the shape of path  $\Pi$ .

We claim that for each  $j$ ,  $\Pi_j$  starts at either  $+30^\circ$  or  $-30^\circ$ . First we show this statement holds for  $j=1$ . Note that the first Delaunay triangle of  $T'$  is to the right of  $A$ ; thus,  $A$  is on the upper left or the lower left of the corresponding empty circle. Therefore, according to step 2 of Algorithm A,  $\Pi_1$ , the piece of  $\Pi$  that starts at vertex  $A$ , starts at an angle of  $\pm 30^\circ$ . For each remaining  $\Pi_j$ , the path produced by

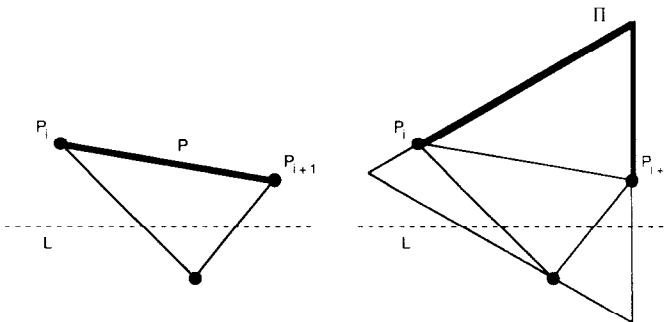
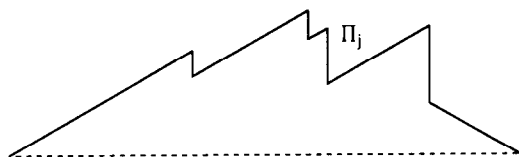


FIG. 10. A portion of  $P$  and the corresponding portion of  $\Pi$ .

FIG. 11. The shape of  $\Pi_j$ .

Algorithm A crosses line  $L$  only when either (1)  $P_i$  is above  $L$  and on the lower left edge of the empty circle  $C$  or (2)  $P_i$  is below  $L$  and on the upper left edge of  $C$ . This is a consequence of choosing the rightmost triangle in step 1, and is enough to conclude that for each  $j$ ,  $\Pi_j$  starts at an angle of  $\pm 30^\circ$ .

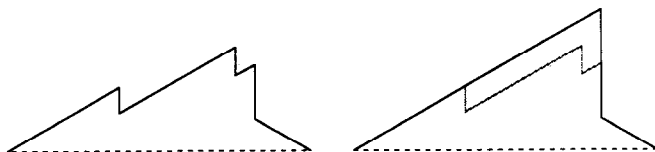
The shape of  $\Pi_j$  is actually even more constrained. The shape is similar for  $\Pi_j$  either above or below  $L$ , so without loss of generality we study the shape of  $\Pi_j$  where  $\Pi_j$  is above  $L$ . From the observations above,  $\Pi_j$  does not cross  $L$  until we get to a vertex that is on the lower left of an empty circle. From the way in which directions are picked (in step 2 of Algorithm A), it follows that, before we get to the  $L$ -crossing segment, only two directions are used in  $\Pi_j$ :  $30^\circ$  and  $-90^\circ$ .

The basic shape of a piece of the path can now be determined.  $\Pi_j$  does some "sawtooth" steps using angles  $30^\circ$  and  $-90^\circ$ , followed by a "tail" at  $-30^\circ$  which crosses  $L$ , as in Fig. 11. A path of this shape can be *unfolded* without affecting its length (see Fig. 12). In this form, it is easy to see that the ratio  $|\Pi_j|/|L_j|$  grows larger as the "tail" grows smaller. Thus, the worst-case ratio appears when the path is the shape of a 30-60-90 triangle. For such a triangle, the ratio is easily seen to be  $(1 + 1/2)/(\sqrt{3}/2) = \sqrt{3}$ . (See Fig. 13.)

In summary, we have shown  $|\Pi_j|/|L_j| \leq \sqrt{3}$ , for each  $j$ . Putting the  $\Pi_j$ 's back together, we get  $|\Pi|/|L| \leq \sqrt{3}$ . By Lemma 1.3,  $|P| \leq |\Pi|$ ; thus  $|P|/|L| \leq \sqrt{3}$ , and the proof of Lemma 1 is complete. ■

What happens if the line between  $A$  and  $B$  is not horizontal? In other words how does one change the proof of Lemma 1 to prove Theorem 1? With some modification, almost the same proof can be used. We start with a restatement of Theorem 1.

**THEOREM 1.** *Let  $S$  be a set of points in the plane and let  $T$  be a TD Delaunay triangulation of  $S$ . For any points  $A$  and  $B$  of  $S$ , there exists an  $A$ -to- $B$  path along edges of  $T$  that has length  $\leq 2|AB|$ .*

FIG. 12. A piece of the path and the same piece *unfolded*.

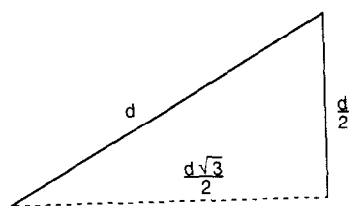


FIG. 13. The worst-case ratio.

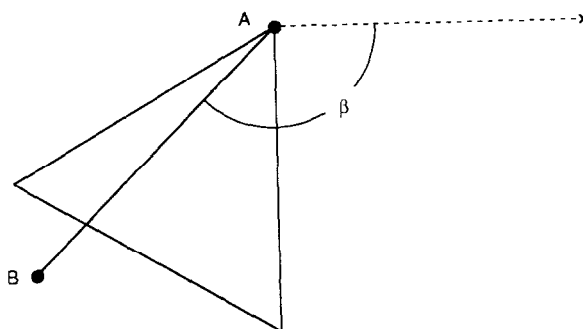
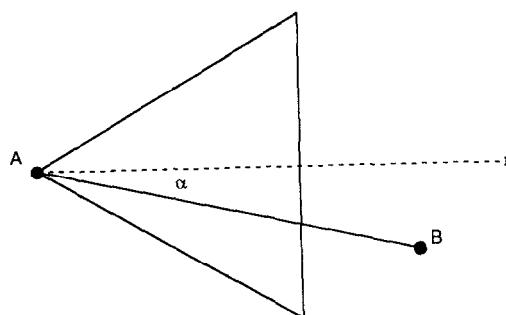
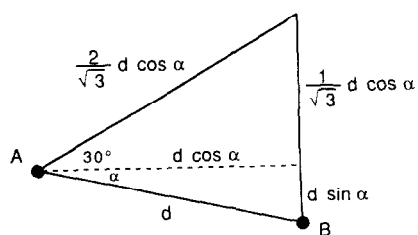

 FIG. 14. An example of how to place an empty circle on segment  $AB$ .

 FIG. 15. Standard orientation of an empty circle placed on  $AB$ .


FIG. 16. The worst-case ratio.

*Proof.* First, take an equilateral triangle with the same orientation as an empty circle and, without changing its orientation, place one of its vertices on either  $A$  or  $B$  in such a way as to make segment  $AB$  go through the interior of the triangle, as in Fig. 14.

To place the triangle correctly on  $A$ , the ray from  $A$  to  $B$  must be at angle  $\beta$  such that  $-30^\circ \leq \beta \leq 30^\circ$ ,  $90^\circ \leq \beta \leq 150^\circ$ , or  $-150^\circ \leq \beta \leq -90^\circ$  (see Fig. 14). This does not cover all the possible orientations, but fortunately the missing orientations can be handled by placing the triangle on  $B$ . Since we can swap the names of  $A$  and  $B$  and rotate the picture if necessary, we can assume, without loss of generality, that the triangle is placed on vertex  $A$  as shown in Fig. 15, where  $\alpha$  is the angle from  $AB$  to the angle bisector of the triangle at vertex  $A$ . By our rule for placing the triangle, the value of  $\alpha$  must be between  $-30^\circ$  and  $30^\circ$ . Lemma 1 covered the case in which  $\alpha = 0^\circ$ .

The remainder of the proof is the same as for Lemma 1 except  $\alpha$  appears as a parameter. For  $\Pi_j$  above  $L$  the ratio  $|\Pi_j|/|L_j|$  is bounded by  $(\sqrt{3}) \cos \alpha + \sin \alpha$ . (See Fig. 16.) Some simple calculus shows that the maximum value for this bound occurs at  $\alpha = 30^\circ$ , where it has the value 2. A similar bound holds for  $\Pi_j$  below  $L$ , completing the proof. ■

### 3. APPLICATIONS

As mentioned in the Introduction, one application is to network design. Networks based on TD Delaunay triangulations have a number of useful properties, including (1) the network is planar, so it requires just  $O(n)$  edges where  $n$  is the number of sites, and (2) the transmission distance between two sites within the network is at most twice the straight line distance between the two sites.

Another application is to motion planning in the plane. Given a source ( $A$ ), a destination ( $B$ ), and a set ( $S$ ) of obstacles, the *motion planning problem* is to determine the best path to move an object from  $A$  to  $B$  without colliding with any of the obstacles. For us, motion is confined to the plane, the object to be moved is a point, and the obstacles are nonoverlapping polygons. Currently, it takes  $O(n^2)$  time, where  $n$  is the number of edges of all the obstacles, to find the shortest  $A$ -to- $B$  path. This method uses the *visibility graph* which can be built in  $O(n^2)$  time [We85, AAGHI85]. Another approach to the motion planning problem is to drop the goal of an optimal path and to look instead for a reasonable path that can be found very efficiently.

The results presented here on TD Delaunay triangulations can be combined with the idea of a *constrained Delaunay triangulation* to produce such a motion planning technique. Intuitively, a *constrained Delaunay triangulation* of the obstacle set  $S$  is a straight-line planar graph  $G$  that is as close as possible to a Delaunay triangulation, but with the restriction that the obstacle edges must be included as edges of  $G$ . This type of triangulation was first introduced by Lee [Le78], where it was called a *generalized Delaunay triangulation*. In [Ch87a], we show that a con-

strained Delaunay triangulation can be built in  $O(n \log n)$  time; similar techniques can be used to show that a constrained TD Delaunay triangulation can also be built in  $O(n \log n)$  time.

A constrained TD Delaunay triangulation,  $G$ , has additional desirable properties. First,  $G$  is a planar graph and as a consequence the shortest  $A$ -to- $B$  path within  $G$  can be found in  $O(n \log n)$  time. Second, as a consequence of Theorem 1, the length of the shortest  $A$ -to- $B$  path *within*  $G$  is at most twice the length of the shortest *possible*  $A$ -to- $B$  path.

These results lead to a technique for finding a reasonably good path, with length at most twice that of the optimal path, in time  $O(n \log n)$ . More information and extensions to this technique appear in [Ch87b].

#### 4. DISCUSSION

Theorem 1 leads to an interesting question: Is the bound of 2 on the ratio of path length to straight-line distance a tight bound? This question can be restated as two separate questions: (1) Is there a better bound for TD Delaunay triangulations? and (2) Is there another type of planar graph with a better bound than 2?

At this point, some additional terminology is useful. For a given graph type  $\tau$ , let  $\tau(S)$  represent a graph of type  $\tau$  built from a set  $S$  of vertices and let  $c(\tau)$  be the smallest number such that for all sets  $S$  of vertices in the plane and for all  $A$  and  $B$  in  $S$ , the length of the  $A$ -to- $B$  path within  $\tau(S)$  is  $\leq c(\tau) |AB|$ . Theorem 1 says  $c(\text{TDDT}) \leq 2$  (TDDT represents *TD Delaunay triangulation*).

The bound of 2 is a tight bound for TD Delaunay triangulations (i.e.,  $c(\text{TDDT}) = 2$ ). Figure 17 shows an example in which the ratio 2 is obtained. The vertices shown are all on the same empty *circle*; thus, they can be triangulated in any fashion to form a TD Delaunay triangulation. (Alternately, the reader can think of moving vertices by some small epsilon to force a particular triangulation.) The straight-line distance between  $A$  and  $B$  is easily seen to be 3, but we can choose

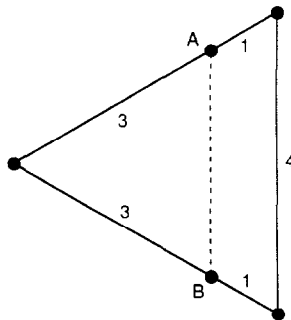


FIG. 17. An example where the  $A$ -to- $B$  path has length twice  $|AB|$ .

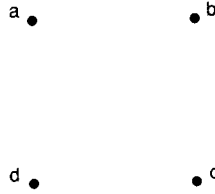


FIG. 18. An example where the ratio is  $\sqrt{2}$  for any planar graph.

a triangulation in which the length of the shortest path following edges of the triangulation is 6.

No type of planar graph is currently known to have  $c(\tau) < 2$ , although the standard Delaunay triangulation may be such a graph type. An attempt to prove that  $c(\text{DT}) = \pi/2$  (DT represents *Delaunay triangulation*) led to the results presented in this paper. Showing  $c(\text{DT}) \geq \pi/2$  is easy, but the other direction is considerably more difficult (in fact, it is not even true).

For the standard Delaunay triangulation,  $\pi/2$  is the best bound possible (i.e.,  $c(\text{DT}) \geq \pi/2$ ). To see this, consider  $A$  and  $B$  at the ends of the diameter of a circle. By adding other points on the circle, we can create Delaunay triangulations for which the ratio of path length to  $|AB|$  is arbitrarily close to  $\pi/2$ .

Recently, Dobkin, Friedman, and Supowit [DFS87] have shown that  $c(\text{DT}) \leq \pi(1 + \sqrt{5})/2$  ( $\approx 5.08$ ). This is the first result to show there is an upper bound on  $c(\text{DT})$ .

Delaunay triangulations seem intuitively appropriate for this kind of result, but there may be other graph types with better characteristics. It is unknown whether there is a non-Delaunay graph type with bound better than 2, but a simple example shows that no *planar* graph type can have a bound better than  $\sqrt{2}$  (i.e., if  $\tau$  is a planar graph type then  $c(\tau) \geq \sqrt{2}$ ). Let  $S = \{a, b, c, d\}$ , where  $a$ ,  $b$ ,  $c$ , and  $d$  are points at the vertices of a unit square as shown in Fig. 18. Since  $\tau(S)$  is planar, it cannot contain both edge  $ac$  and edge  $bd$ . Without loss of generality, we assume  $\tau(S)$  does not contain  $ac$ . It is easy to see that  $|ac| = \sqrt{2}$  and that the length of the  $a$ -to- $c$  path within  $\tau(S)$  is  $\geq 2$ .

## 5. CONCLUSIONS

This paper and an earlier, conference paper [Ch86] introduce a property of planar graphs that has apparently never before been studied. There exists a constant  $c$  such that for any set  $S$  of points in the plane there is a planar graph  $G$  with the property that for any points  $A$  and  $B$  of  $S$ , the length of the  $A$ -to- $B$  path in  $G$  is  $\leq c|AB|$ . For a given graph type we are interested in the smallest such constant  $c$ .

Types of graphs known to have this property are all variations of Delaunay

triangulations. For the standard Delaunay triangulation,  $c$  is less than about 5.08 [DFS87]; for the  $L_1$  Delaunay triangulation,  $c \leq \sqrt{10}$  [Ch86]; and, as shown in this paper, for the TD (triangle distance) Delaunay triangulation,  $c = 2$ . For the standard Delaunay triangulation, I suspect  $c$  is close to  $\pi/2$ .

I expect these results to have a wide range of applications. Early applications, outlined in Section 3, include network design and motion planning. These applications take advantage of the fact that the graphs are planar, allowing compact data structures and fast running times.

## REFERENCES

- [AAGHI85] T. ASANO, T. ASANO, L. GUIBAS, J. HERSHBERGER, AND H. IMAI, Visibility-polygon search and Euclidean shortest paths, in "Proceedings, 26th IEEE Symposium on Foundations of Computer Science, 1985," pp. 155–164.
- [CD85] L. P. CHEW AND R. L. DRYSDALE, Voronoi diagrams based on convex distance functions, in "Proceedings, 1st Symposium on Computational Geometry, Baltimore, 1985," pp. 235–244; revised version, *Discrete Comput. Geom.*, in press.
- [Ch86] L. P. CHEW, There is a planar graph almost as good as the complete graph, in "Proceedings, Second Annual Symposium on Computational Geometry, 1986," pp. 169–177.
- [CH87a] L. P. CHEW, Constrained Delaunay triangulations, "Proceedings, Third Annual Symposium on Computational Geometry, 1987," pp. 215–222; revised version, *Algorithmica* 4 (1989), 97–108.
- [Ch87b] L. P. CHEW, Planar graphs and sparse graphs for efficient motion planning in the plane, manuscript.
- [DFS87] D. P. DOBKIN, S. J. FRIEDMAN, AND K. J. SUPOWIT, Delaunay graphs are almost as good as complete graphs, in "Proceedings, 28th IEEE Symposium on Foundations of Computer Science, 1987," pp. 20–26.
- [Fo87] S. FORTUNE, A sweepline algorithm for Voronoi diagrams, *Algorithmica* 2 (1987), 153–174.
- [Le78] D. T. LEE, "Proximity and Reachability in the Plane," Technical Report R-831, Coordinated Science Laboratory, University of Illinois, 1978.
- [LS80] D. T. LEE AND B. SCHACHTER, Two algorithms for constructing Delaunay triangulations, *Internat. J. Comput. Inform. Sci.* 9, No. 3 (1980), 219–242.
- [PS85] F. P. PREPARATA AND M. I. SHAMOS, "Computational Geometry," Springer-Verlag, New York/Berlin, 1985.
- [We85] E. WELZL, Constructing the visibility graph for  $n$  line segments in  $O(n^2)$  time, *Inform. Process. Lett.* 20 (1985), 167–171.