Discrete Comput Geom 10:197–213 (1993)

Discrete & Computational
Geometry

An Upper Bound for Conforming Delaunay Triangulations*

Herbert Edelsbrunner[1] and Tiow Seng Tan[2]

[1] Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, IL 61801, USA

[2] Department of Information Systems and Computer Science, National University of Singapore, Republic of Singapore

Abstract. A plane geometric graph $\mathscr{C}$ in $\mathbb{R}^2$ *conforms* to another such graph $\mathscr{G}$ if each edge of $\mathscr{G}$ is the union of some edges of $\mathscr{C}$. It is proved that, for every $\mathscr{G}$ with n vertices and m edges, there is a completion of a Delaunay triangulation of $O(m^2n)$ points that conforms to $\mathscr{G}$. The algorithm that constructs the points is also described.

1. Introduction

Decompositions of two- and higher-dimensional domains play a major role in many engineering applications. For example, the finite element analysis method is based on the decomposition of a domain into so-called *elements* [StF]. A particularly important class of decompositions are simplicial cell complexes, sometimes referred to as *triangulations*. Here the domain is decomposed into simplices (triangles in two and tetrahedra in three dimensions) so that the intersection of two simplices is either empty or a face of both. Applications of triangulations can be found in finite element analysis [C], surface interpolation [La], shape reconstruction [Bo], and other research areas.

An important type of triangulation is the Delaunay triangulation [De]. It is dual to the so-called Voronoi diagram [V]. The popularity of the two-dimensional Delaunay triangulation is partly due to the fact that it optimizes various quality measures, including the smallest angle [Si], the largest circumscribed circle [D'AS],

* Research of the first author is supported by the National Science Foundation under Grant CCR-8921421 and under the Alan T. Waterman award, Grant CCR-9118874. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the National Science Foundation. Work of the second author was conducted while he was on study leave at the University of Illinois.

the largest minimum enclosing circle [D'AS], [Ra1], and the integral of the gradient squares [Ri]. Algorithms that construct the Delaunay triangulation of a given set of n points in the plane in time $O(n \log n)$ can be found in [GS], [F], [GKS], and other publications in computational geometry [PS], [Ed].

In practical applications it is often the case that a constraining set of points and line segments must be part of the triangulation. The constraining set is most naturally modeled as a geometric graph whose vertices are the points and endpoints and whose edges are the line segments in the set. It is assumed that the line segments intersect at most at endpoints.

Of course, the Delaunay triangulation of the vertices of such a geometric graph will, in general, not contain all edges of the graph. This leads to the definitions of constrained and of conforming Delaunay triangulations. The *constrained* Delaunay triangulation of a geometric graph is, in some sense, the best approximation of the Delaunay triangulation given that it must contain the graph [LL]. For a graph with n vertices it can be constructed in time $O(n \log n)$, see, e.g., [Se]. A *conforming* Delaunay triangulation is a genuine Delaunay triangulation, or, more precisely, a completion of a degenerate Delaunay triangulation. Its relation to the constraining graph is that each vertex of the graph is also a vertex of the triangulation, and each edge of the graph is the union of edges of the triangulation [BFL]. Constructing a conforming Delaunay triangulation of a geometric graph is usually harder than constructing the constrained Delaunay triangulation. The reason is that the graph can force the introduction of a large number of points to achieve conformity. Previous work [Bo], [NS], [O], [Sa], [SaP] fails to provide upper bounds on the number of points necessary for a conforming Delaunay triangulation that are polynomial in the size of the constraining graph. Such a bound is given in this paper.

The paper is organized as follows. Section 2 formalizes the problem and Section 3 presents some preliminary results. Section 4 proves the upper bound on the number of points needed for a conforming Delaunay triangulation. Section 5 explicitly formulates the algorithm that is implicit in the proof of the upper bound. Section 6 concludes the paper with some open problems.

2. The Problem Definition

First some notation. Let V be a set of n points in $\mathbb{R}^2$. An *edge* is a closed line segment connecting two points. Let E be a collection of edges. Then $\mathcal{G} = (V, E)$ is a *plane geometric graph* if

(i) no edge contains a vertex other than its endpoints, that is, $ab \cap V = \{a, b\}$ for every edge $ab \in E$, and
(ii) no two edges *cross*, that is, $ab \cap cd \in \{a, b\}$ for every two edges $ab \neq cd$ in E.

The connected components of $\mathbb{R}^2$ minus all vertices of V and all points on edges of E are the *faces* of $\mathcal{G}$. For example, if the edges in E are pairwise disjoint, then $\mathcal{G}$ is a matching and there is only one face. Another common case is when E and V form a single cycle. This cycle is the boundary of a

polygon, and $\mathcal{G}$ has two faces, the inside and the outside of the polygon. If V is fixed and E is maximal so that no two edges cross, then we call $\mathcal{G}$ a *triangulation.* In this case all bounded faces are triangles and their union is the convex hull of V.

An edge ab, $a, b \in V$, is a *Delaunay edge* if there is a circle through a and b so that all other points of V lie outside the circle. The collection of Delaunay edges defines a plane geometric graph $\mathcal{D}(V)$ known as the *Delaunay triangulation* of V. In the nondegenerate case, which excludes four or more points on a common circle, $\mathcal{D}(V)$ is indeed a triangulation. Even in degenerate cases the faces of $\mathcal{D}(V)$ are convex polygons, and these can be further subdivided into triangles using additional edges. Each additional edge cd, $c, d \in V$, has the property that there is a circle through c and d so that all other points of V lie *on* or outside this circle. The resulting triangulation is called a *completion* of $\mathcal{D}(V)$.

The problem studied in this paper can now be described as follows. Let $\mathcal{G} = (V, E)$ be a plane geometric graph. A completion $\mathcal{C}$ of a Delaunay triangulation *conforms* to $\mathcal{G}$ if every vertex of $\mathcal{G}$ is a vertex of $\mathcal{C}$ and every edge of $\mathcal{G}$ is the union of edges of $\mathcal{C}$. The problem is to find a small point set S so that $\mathcal{D}(S)$ has a completion that conforms to $\mathcal{G}$. We call such a completion a *conforming Delaunay triangulation* of $\mathcal{G}$. It is also desirable to have an algorithm that constructs S as well as a completion of $\mathcal{D}(S)$ that conforms to $\mathcal{G}$. The remainder of this section shows that the latter task can be handled by existing constrained Delaunay triangulation algorithms [Se], once we have an algorithm that finds the points.

As mentioned above, each edge ab of a completion of $\mathcal{D}(S)$ satisfies the *empty open disk property*, that is, there exists a circle through a and b so that no point of S belongs to the open disk bounded by the circle. We now argue that this property is also sufficient for the existence of a conforming Delaunay triangulation. Call the closed portion of an edge of $\mathcal{G}$ between two contiguous points of S on this edge an *interval.*

2.1. *$\mathcal{D}(S)$ has a completion that conforms to $\mathcal{G}$ iff every interval defined by $\mathcal{G}$ and S has the empty open disk property with respect to S.*

Proof. The only if part follows from the definition of a completion of $\mathcal{D}(S)$. For the if part assume that every interval ab has the empty open disk property. If ab is an edge of $\mathcal{D}(S)$, then nothing has to be proved. Otherwise, no edge of $\mathcal{D}(S)$ can cross ab because every circle passing through the endpoints of such an edge encloses a or b or both. Because $\mathcal{G}$ is plane, there is also no other interval that crosses ab. So ab, and all other intervals that are not edges of $\mathcal{D}(S)$, can be added to $\mathcal{D}(S)$ without introducing any crossing. We can add zero or more noncrossing edges arbitrarily until a completion of $\mathcal{D}(S)$ is obtained. □

3. Preliminary Results

An edge $ab \in E$ that belongs to the boundary of the convex hull of V automatically satisfies the empty open disk property, and no points on ab need to be introduced.

Fig. 3.1. An $\Omega(mn)$ lower bound example on the number of vertices of a conforming Delaunay triangulation. In the example shown, $m = 6$, $k = 3$, and therefore $n = 2m + 2k = 18$.

For other edges ab there are points on both sides of the line that contains ab. It is thus possible that ab does not satisfy the empty open disk property, in which case points must be added to subdivide ab into smaller intervals.

In some cases the size of S must be at least quadratic in the size of $\mathcal{G}$. This bound can be shown using the example of Fig. 3.1 which consists of $m = |E|$ edges and $n = |V| = 2m + 2k$ vertices. The edges are parallel and very close to each other. The isolated vertices come in k pairs, with one vertex on each side of the group of edges. Provided the edges are sufficiently close to each other, and the vertices are sufficiently close to the edges, it will be necessary to place a point approximately between the two vertices of every pair on every edge. This proves that at least mk points need to be added to obtain a conforming Delaunay triangulation. The lower bound of $\Omega(mn)$ follows for $k = \Omega(n)$. For smaller k, the endpoints of half of the m edges can be used to play the role of the isolated vertices.

A common approach to produce a conforming Delaunay triangulation is to place sufficiently many points on the edges of the constraining graph so that each interval has a circle that avoids all other edges [Bo], [NS], [O], [Sa], [SaP]. This can always be achieved except maybe at places close to shared endpoints where sharp angles are formed. This special case is handled by placing points at the intersections of the edges with a sufficiently small circle drawn around the common endpoint. The method avoids the need for backtracking as no point placed on any edge harms any interval that already has such a circle. The price, however, is a possibly horrendous number of new points. Indeed, there is no function $f(n)$ that can bound the number of points although for every problem instance it is finite. In particular, the number of points added grows as the edges move closer to each other.

An upper bound that depends solely on n can be otained as follows. Initially, set $S = V$ and consider all m edges as *unprotected*. Treat the edges of $\mathcal{G}$ in turn. At the time the ith edge is treated, it may consist of various protected and unprotected intervals. Place sufficiently many points on the unprotected intervals so that each new interval has a circle that does not enclose any point of the current set S. Each such circle may, however, intersect other edges. To prevent points from being placed inside the circle later in the process, we place points at the intersections between the circle and any unprotected interval of another edge. Declare each new interval as protected if it is enclosed by the circle and unprotected otherwise. See Fig. 3.2. The number of points needed to treat the ith edge does not exceed the current size of S since it suffices to project the current set S orthogonally onto the ith edge. Similarly, the number of circles needed for the ith edge does not exceed the current size of S. Assume inductively that $|S| \leq n(2m + 1)^{i-1}$ before the next step that treats the ith edge. The next step creates at

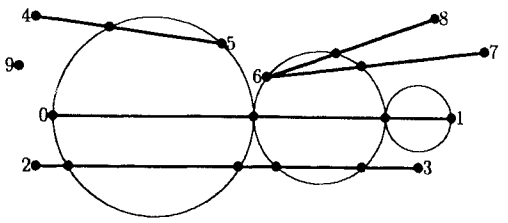

Fig. 3.2. The original graph, $\mathcal{G}$, has vertices 0–9 and edges 01, 23, 45, 67, and 68. First, edge 01 is treated. In the process, two new points are added to 01. The resulting three circles cross the other edges in seven points, which are also added. Each edge of $\mathcal{G}$ now consists of protected and unprotected intervals.

most $n(2m + 1)^{i-1}$ circles intersecting the remaining edges in at most $2m$ points each. The size of S thus increases to at most

$$n(2m + 1)^{i-1} + n(2m + 1)^{i-1}2m = n(2m + 1)^i.$$

The total number of points at the end of the process is therefore at most $n(2m + 1)^m$.

This method apparently produces far too many points. An improvement was found by Mehlhorn, Sharir, and Welzl. Their method combines the projection of points with a divide-and-conquer scheme and achieves a subexponential although not yet polynomial bound. The idea of protected and unprotected intervals turns out to be valuable in our effort to obtain a polynomial upper bound on the number of points.

4. The Upper Bound

Given a plane geometric graph $\mathcal{G} = (V, E)$, with $|V| = n$ and $|E| = m$, this section shows how to find $O(m^2n)$ points so that each resulting interval has the empty open disk property. As defined earlier, an interval is the closed portion of an edge between two contiguous points of S chosen on this edge. If no point of an edge belongs to S, except its endpoints, then this edge itself is an interval.

The Global Idea. The point set S is constructed in two steps, the *blocking* and the *propagation phase.* Initially, S contains only the vertices of $\mathcal{G}$, that is, $S = V$.

The goal of the blocking phase is to find $O(n)$ pairwise disjoint open disks that contain no points of V so that the union of their closures is connected and contains V. Each circle bounding such a disk is called a *blocking circle.* After finding these disks, we add the intersections between blocking circles and edges of $\mathcal{G}$ to the set S. In addition, we add the $O(n)$ points at which blocking circles touch each other. The new set S forms the vertex set W of a plane geometric graph $\mathcal{H}$ which conforms

to $\mathscr{G}$. The edges of $\mathscr{H}$ are the intervals of $\mathscr{G}$ together with edges connecting contiguous points of S on blocking circles.

$\mathscr{H}$ has two types of edges. Each *protected* edge is enclosed by a blocking circle; its endpoints lie on the blocking circle. All other edges are *unprotected.* By construction, protected edges have the empty open disk property with respect to the current set S. We make sure that no points inside the blocking circles are added to S later so that this property persists with respect to all future sets S.

The unprotected edges are further subdivided into intervals in the propagation phase. For an edge or interval ab, we define the *minidisk* of ab, M_{ab}, as the smallest open disk whose closure contains ab. If ab is unprotected and its minidisk contains a point $c \in S$ visible from every point of ab, then a point c' subdividing ab into ac' and $c'b$ is added to S. This point c' is chosen so that c lies outside $M_{ac'}$ and $M_{c'b}$.

The Blocking Phase. We show how to use a minimum spanning tree of V to construct $n-1$ open disks $D_1, D_2, \ldots, D_{n-1}$ that satisfy the following properties:

(1) $D_i \cap V = \varnothing$ for all $1 \leq i \leq n-1$,
(2) $D_i \cap D_j = \varnothing$ for all $1 \leq i < j \leq n-1$,
(3) $D = \bigcup_{i=1}^{n-1}$ (closure of D_i) is connected, and
(4) $V \subseteq D$.

A *minimum spanning tree* $\mathscr{T}$ of V is a spanning tree of the complete geometric graph $\left(V, \binom{V}{2}\right)$ whose sum of edge lengths is a minimum. An important property of $\mathscr{T}$ is that the minidisk of every edge of $\mathscr{T}$ is disjoint from V, see, e.g., Section 13.2.5 of [Ed].

Label the vertices of $\mathscr{T}$ (the points of V) from 0 through $n-1$ so that, for every $0 \leq j \leq n-1$, the vertices $0, 1, \ldots, j$ induce a subtree of $\mathscr{T}$. Define i_j so that $i_j j$ is an edge of this subtree. Notice that $i_j < j$ and that i_j is unique. The edges $i_j j$ are now used to define the disks D_j. The disk D_1 is the minidisk of edge 01. The disk D_j, for $2 \leq j \leq n-1$, is maximal so that

(i) its center lies on $i_j j$,
(ii) its bounding circle goes through j, and
(iii) it is disjoint from disks D_1 through D_{j-1} constructed earlier.

Clearly, $D_j \subseteq M_{i_j j}$. This implies property (1). Properties (2), (3), and (4) follow from the construction.

Let V' be the set of points where the blocking circles intersect the edges of $\mathscr{G}$, and let V'' be the set of points (not in V') where the blocking circles touch each other. As described above, the points in V' and V'' are added to S. Let W be the new set S and let $\mathscr{H} = (W, F)$ be a plane geometric graph with $F = F' \cup F''$ defined as follows. The set F' contains all intervals on edges of $\mathscr{G}$. Remember that by construction all points of W lie on the $n-1$ blocking circles. Consider an open disk D_i bounded by a blocking circle C_i and let $p_0, p_1, \ldots, p_{k-1}, p_k = p_0$ be the points of W that lie on C_i in this sequence. These points define a convex k-gon

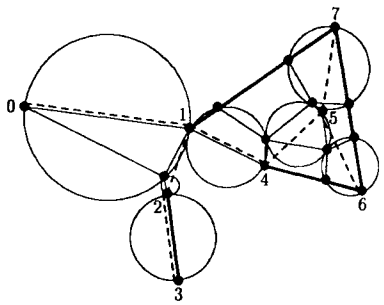

Fig. 4.1. The original graph, $\mathcal{G}$, has vertices 0–7 and edges 17, 23, 46, and 67. The edges of the minimum spanning tree, 01, 12, 23, 14, 45, 56, and 57, are indicated by broken lines. Each edge of the tree corresponds to a blocking circle. Each blocking circle encloses protected edges of $\mathcal{H}$.

with edges $p_l p_{l+1}$, $0 \leq l \leq k-1$, termed *walls*. Some of these walls may be intervals on edges of $\mathcal{G}$ and therefore belong to F'. In any case, F'' is the collection of all walls. This completes the definition of $\mathcal{H}$ which conforms to $\mathcal{G}$, see Fig. 4.1.

Note that each wall is protected by a blocking circle. The collection of walls defines another plane geometric graph, $\mathcal{I} = (W, F'')$. Clearly, $\mathcal{I}$ is a subgraph of $\mathcal{H}$. It is convenient to adopt the topology of the sphere. In this model all faces of $\mathcal{H}$, including the outside face, are simply connected because $\mathcal{H}$ is connected. Similarly, all faces of $\mathcal{I}$ are simply connected.

The Propagation Phase. The unprotected edges of $\mathcal{H}$ are further subdivided into intervals during a nondeterministic process. Initially, every unprotected edge is also an unprotected interval. Consider an unprotected interval ab and its minidisk M_{ab}. Call a point $c \in S$ *visible* from ab if the open triangle abc is disjoint from all edges of $\mathcal{H}$. Suppose a point $c \in S$ visible from ab is contained in M_{ab}. We add c' to S, where c' is the orthogonal projection of c onto ab, and thus subdivide ab into ac' and $c'b$. Repeat this step until there is no unprotected interval ab with such a point c.

This completes the description of how the point set S is constructed. The remainder of this section shows that the eventual size of S is $O(m^2 n)$. The blocking phase adds at most $(2m-1)(n-1)$ intersection points between edges and circles and fewer than $3n$ points where circles touch each other. The latter bound follows from the planarity of the intersection graph of the blocking circles.

We now focus on proving that each point created in the blocking phase gives rise to at most $O(m)$ points in the propagation phase. We begin by proving a few properties of the propagation process. Let ab be an unprotected interval at some point in time during the process, and let $c \in S$ lie in M_{ab}. Then the orthogonal projection c' of c onto the line through a and b lies strictly between a and b. Furthermore, $c \notin M_{ac'}$ and also $c \notin M_{c'b}$.

Assume now that $c \neq x, y$ is a point on some edge xy of $\mathcal{G}$. All points in $S - (V \cup V'')$ are of this form. Then we have the following property:

4.1. There are at most two intervals, ab and $a'b'$, so that c is visible from both and contained in their minidisks. Furthermore, ab and $a'b'$ lie on different sides of the line through x and y.

Property 4.1 holds because if $c \in M_{ab}$, then $\angle acb > \pi/2$, and if c is visible from ab, then a and b lie on the same side of the line through x and y. The same is true for $a'b'$. So if ab and $a'b'$ lie on the same side of the line, then one endpoint of $a'b'$ must lie between a and b as seen from c. This contradicts the assumption that c is visible from ab and from $a'b'$.

Let $c \in M_{ab}$ be visible from ab, and let c' be the orthogonal projection of c onto ab. Then we have the following property:

4.2. There is no interval $a''b''$ on the same side of the line through a and b as c so that c' is visible from $a''b''$ and $c' \in M_{a''b''}$.

Assume such an $a''b''$ exists. Then $\angle a''c'b'' > \pi/2$ which implies that c lies between a'' and b'' as seen from c'. This either contradicts that c is visible from ab or that c' is visible from $a''b''$.

Assume now that $c \in S$ does not lie strictly between the endpoints of an edge of $\mathscr{G}$, so $c \in V \cup V''$. Similar to property 4.1 we have the following property:

4.3. There are at most three intervals ab so that c is visible from ab and $c \in M_{ab}$.

The reason for property 4.3 is simply that at most three angles larger than $\pi/2$ can be packed around c.

A Locality Property. Notice that the propagation phase takes care only of local constraints. In other words, it considers only visible point–interval pairs c, ab. Although the minidisk of ab can contain other points of S, it is indeed justified to ignore such points, as we will see shortly. Let ab be an unprotected interval. We call the minidisk M_{ab} *locally empty* if it contains no point of S that is visible from ab. Furthermore, M_{ab} is *empty* if it contains no point of S at all.

To prepare for the next lemma we consider an interval ab and a point $c \in M_{ab}$. If c is not visible from ab, then there are intervals st that intersect the open triangle abc. We say that st *separates* c from ab if both endpoints, s and t, lie outside M_{ab}. Otherwise, st *hinders* the visibility between c and ab but it does not separate. Let $E_{c,ab}$ be the set of intervals that separate c and ab, and define $F_{c,ab}$ as the set of (nonseparating) intervals that hinder the visibility between c and ab. It is interesting to observe that $F_{c,ab} = \emptyset$ or there is another point d of S in M_{ab} with $E_{d,ab} \subseteq E_{c,ab}$ and $F_{d,ab} = \emptyset$. To see this, choose a point $x \in abc$ so that the open triangle abx does not intersect any edge of $\mathscr{H}$. Move x continuously and straight toward c until either a side of the triangle abx hits a vertex d or x hits a nonseparating interval uv of $\mathscr{H}$. In the second case at least one of the two endpoints, say u, lies in M_{ab}. Slide x on uv toward u until either a side of the triangle abx hits another

vertex d or x reaches u (then $d = u$), whichever happens first. The path of x crosses only intervals that separate c and ab, therefore $E_{d,ab} \subseteq E_{c,ab}$.

4.4. *If the minidisks of all unprotected intervals are locally empty, then they are all empty.*

Proof. Suppose the claim is false. Then there is an unprotected interval ab whose minidisk M_{ab} contains a point $c \in S$. As argued in the preceding paragraph, we can make the extremum assumption that ab and c are chosen so that the number of separating intervals $E_{c,ab}$ is a global minimum and $F_{c,ab} = \emptyset$. Let st be an interval that separates ab from c. Note that st is not protected because every circle through s and t encloses at least one of a, b, and c. However, if st is an unprotected interval, we have $c \in M_{st}$ because s and t are outside M_{ab}. Furthermore, st cuts M_{ab} into two pieces, and the piece that contains c is properly contained in M_{st}. Therefore, $E_{c,st} \subseteq E_{c,ab}$ and we have proper containment because $st \in E_{c,ab}$ does not belong to $E_{c,st}$. This either contradicts the extremum assumption or that st is locally empty. □

Propagation Sequences. We are now ready to analyze the number of points created in the propagation phase. Our particular goal is to show that each point c created in the blocking phase generates at most $3m$ points in the propagation phase. We say that a point c *generates* another point d if there is a sequence $c = c_0, c_1, \ldots, c_k = d$ so that c_{i+1} is created as the orthogonal projection of c_i onto some interval during the propagation phase, for $0 \leq i \leq k - 1$. The sequence $c_0, c_1, \ldots, c_k$ is called a *propagation sequence*. It is *nontrivial* if $k \geq 1$, and it is *maximal* if c_0 is created in the blocking phase and c_k generates no further point. Note that all c_i, $i \geq 1$, of a maximal propagation sequence lie on unprotected edges of $\mathcal{H}$, and these edges are contained in edges of $\mathcal{G}$.

Every point d created in the propagation phase gives rise to at most one point d'. To see this note that such a point d lies strictly between the endpoints of an edge xy of $\mathcal{G}$. Now property 4.1 implies that d is visible from at most two intervals whose minidisks contain d. By property 4.2 and because d itself is generated by orthogonal projection, d generates another point d' on at most one of these two intervals. Together with property 4.3, this implies that a point of S constructed before the propagation phase gives rise to at most three nontrivial maximal propagation sequences. In fact, there are at most two such sequences per point not in $V \cup V''$. To establish that the length of a maximal propagation sequence is at most m, it suffices to prove the following:

4.5. *No propagation sequence can have two or more points on the same edge of* $\mathcal{G}$.

Proof. Suppose the claim is false. Consider a minimal propagation sequence, $c = c_0, c_1, \ldots, c_k = d$, so that c and d lie on the same edge xy of $\mathcal{G}$. Consider the polygon, P, whose boundary consists of the line segments cd and c_ic_{i+1}, for

$0 \leq i \leq k-1$. Recall that walls are protected, so no points are projected on or across them. Thus, P lies completely within a face of $\mathscr{I}$. Since all vertices of $\mathscr{G}$ are also vertices of $\mathscr{I}$, and because all faces of $\mathscr{I}$ are simply connected, there can be no vertex of $\mathscr{G}$ inside P. Thus, each edge of $\mathscr{G}$ that intersects P has its endpoints outside P. It thus intersects the boundary of P in at least two points. Since c_0, $c_1, \ldots, c_k$ is minimal, it follows that $k = 2$ and that c_0 and c_2 lie on the same side of the edge xy of $\mathscr{G}$ that contains c_1. However, this contradicts property 4.2. □

As mentioned earlier, $|V \cup V''| < 4n$. Each point $c \in V \cup V''$ gives rise to at most three maximal propagation sequences of length at most $m + 1$ each. The point c itself is the only point of these sequences that does not necessarily lie on an edge of $\mathscr{G}$. The number of other points created during the blocking phase is less than $2mn$. Each such point gives rise to at most two maximal propagation sequences of length at most m each. The total number of points after the propagation phase is thus less than

$$4n(3m + 1) + 2mn(2m - 1) = 4m^2n + 10mn + 4n.$$

This proves the main result of this paper.

Theorem 4.6. *Let $\mathscr{G} = (V, E)$ be a plane geometric graph with $|V| = n$ and $|E| = m \geq 1$. There exists a point set S of size $|S| = O(m^2n)$ so that its Delaunay triangulation has a completion that conforms to $\mathscr{G}$.*

5. Implementing the Proof

The proof of the $O(m^2n)$ upper bound presented in Section 4 is constructive and can be translated into an algorithm without much effort. The only demanding step is the implementation of the propagation phase. In order to keep the time-complexity roughly within the same order of magnitude as the number of points added, we need to project the points in a sequence that is computationally inexpensive. We assume that point coordinates can be stored in a constant amount of storage and that basic geometric operations, such as intersecting a circle with an edge and projecting a point onto a line, can be carried out in a constant amount of time.

The Blocking Phase. A minimum spanning tree of a set of n points in the plane can be computed in time $O(n \log n)$, see, e.g., Section 13.2.5 of [Ed]. This requires the construction of the Delaunay triangulation of the points and running a standard minimum spanning tree algorithm on the geometric graph of this triangulation, see, e.g., [CLR]. Alternatively, a minimum spanning tree can be obtained in time $O(n^2)$ directly from the complete geometric graph of the points. The slower method is certainly easier to implement.

After computing the tree, we need to find the open disks $D_1, D_2, \ldots, D_{n-1}$ that satisfy properties (1)–(4). Most straightforwardly, these disks can be constructed one by one as explained in Section 4. For each j, the largest disk D_j needs to be found so that its center lies on $i_j j$, j lies on the bounding circle of D_j, and D_j avoids all D_i, $i < j$. This can be done in time $O(j)$. The total amount of time for this step is thus $O(n^2)$. The plane graphs $\mathscr{H}$ and $\mathscr{I}$ can be computed by intersecting the bounding circles of the D_j with each other and with the edges of $\mathscr{G}$. The resulting $O(mn)$ intersection points can be computed and sorted along circles and edges in time $O(mn \log n)$.

A Tree of Regions. Recall that $\mathscr{I}$ is a subgraph of $\mathscr{H}$ and contains none of its unprotected edges. A point $c \in S$ is projected onto an edge ab only if ab is unprotected. Thus, projections happen only within faces of $\mathscr{I}$. As mentioned earlier, $\mathscr{I}$ is connected and therefore its faces are simply connected. We can thus restrict our attention to a single face of $\mathscr{I}$.

Let f be a face of $\mathscr{I}$ that is further subdivided into *regions* by unprotected edges of $\mathscr{H}$. Let $\mathscr{R}_f$ be the graph whose nodes are the regions of f, and whose arcs connect regions that share unprotected edges of $\mathscr{H}$. Each subdividing unprotected edge has both endpoints on the boundary of f, which implies that $\mathscr{R}_f$ is a free tree. It is convenient to fix an arbitrary node as its root and thus impose a parent–child relation on adjacent node pairs. Points are projected onto unprotected edges in three stages. The first stage computes an initial set of projections that avoids difficult situations in the second stage. The second stage computes and sorts segments along the boundary of each region. The last stage consists of a postorder and a preorder traversal of $\mathscr{R}_f$.

Consider two adjacent nodes μ and ν of $\mathscr{R}_f$, and let ab be their shared unprotected edge. Points on the boundary of μ that are projected onto ab are said to be *exported* from μ to ν. Symmetrically, we say they are *imported* by ν. The points projected onto ab are stored in two separate sorted lists, $L_{\mu\nu}$ and $L_{\nu\mu}$, one for each side of ab. The complete list of points exported from μ to ν, $L_{\mu\nu}$, can be computed only after all import lists $L_{\kappa\mu}$, for $\kappa \neq \nu$ adjacent to μ, are available. note that the import list from ν, $L_{\nu\mu}$, is not necessary for computing $L_{\mu\nu}$ because a propagation sequence follows only one direction of a path in $\mathscr{R}_f$ (see property 4.5).

Stage 1: Subdividing Unprotected Edges. A vertex of μ is *reflex* if the angle inside μ exceeds π. The first stage projects every reflex vertex c onto all unprotected edges ab for which there exists a portion $a'b' \subseteq ab$ so that c is visible from $a'b'$ and $c \in M_{a'b'}$. Since $\angle a'cb' > \pi/2$ there can be at most three such edges ab. As a precaution, we do not require that c be visible from ab. This way c does not need to be reconsidered after ab gets subdivided. Although such projections are not prescribed by the proof in Section 4, they neither invalidate the correctness nor the analysis of the construction. Each reflex vertex is necessarily a vertex in $V \cup V''$, so there are fewer than $4n$ of them. Since each vertex is projected at most three times, we thus increase the number of unprotected edges by less than $12n$.

Here is how we find the at most three unprotected edges for a reflex vertex c. The part of μ visible from c can be computed in a single walk along the boundary of μ, see, e.g., [ElA], [Le], [JS]. The amount of time needed for the walk is proportional to the number of edges. Select the at most three edges that have connected portions visible from c along an angle exceeding $\pi/2$. Project c orthogonally onto these at most three edges. Each projection subdivides an unprotected edge into two such edges.

The rest of the algorithm uses the subdivision of $\mathscr{H}$ produced in stage 1. It is therefore convenient to call the elements of this subdivision vertices and edges. After the completion of stage 1, no reflex vertex visible from an unprotected edge ab lies inside the minidisk M_{ab}. It thus follows that if a point c lies in M_{ab} and is visible from a point on ab, then it is also visible from ab.

Stage 2: Computing Boundary Segments. This stage is a preprocessing step that speeds up computations in stage 3. It prepares the boundary of μ in such a way that points can be projected onto various unprotected edges in a single walk along the boundary of μ. To this end we associate pieces of the boundary, called *segments*, with the unprotected edges of μ. A *segment* for an unprotected edge ab is a maximal connected piece of μ's boundary so that every point x of the segment is visible from ab and contained in M_{ab}. Note that segments do not include their endpoints. Since μ is simply connected, a point x is visible from ab iff it is visible from a and also from b. The segments of ab are constructed as follows:

1. Find the part of μ's boundary visible from a. As mentioned above, this can be done in a single walk along the boundary of μ.
2. Find the part of μ's boundary that is also visible from b. Again a single walk suffices.
3. Intersect the identified boundary pieces with M_{ab}.

Define the *rank* of μ, $r(\mu)$, equal to the number of unprotected edges of μ, and let $|\mu|$ be the total number of edges of μ. If an edge bounds μ on both sides, that is, the edge belongs to the interior of the closure of μ, then it is counted twice. By construction, this can be the case only for protected edges. After carrying out steps 1–3 for each unprotected edge of μ, we obtain a collection of segments. Because of property 4.1 the segments along the boundary of μ are pairwise disjoint. In other words, the segments form a sequence, and they can be sorted in time proportional to $|\mu|$ plus their number. This is because along an edge of μ the segments are ordered consistently with the order of their corresponding unprotected edges. We see later that the number of segments is less than $2r(\mu) + |\mu|$.

Stage 3: Traversing the Tree. In a postorder traversal, the children κ of a node μ are visited before μ. Visiting a node μ here means computing the export list to its parent ν. Notice that because of stage 1, μ and ν may share several unprotected edges. Still, their union is the original unprotected edge of μ and ν, and $L_{\mu\nu}$ can be obtained by concatenating the lists obtained by projecting points onto these unprotected edges. We can assume that, at the time the export list of μ to ν is computed, all import lists $L_{\kappa\mu}$ are available.

List $L_{\mu\nu}$ is constructed in a single walk along the boundary of μ. Whenever a segment that belongs to an unprotected edge shared by μ and ν is encountered, the points on this segment are projected orthogonally onto the unprotected edge. These points can be vertices of μ or points in import lists of μ. The result is the list $L_{\mu\nu}$. It is automatically sorted if we process the points in their order along the boundary of μ.

After the postorder traversal of $\mathscr{R}_f$, all child-to-parent lists are complete. In order to compute the export lists of a node μ to its children κ, we first need to construct the import list from its parent, $L_{\nu\mu}$. This is done in a preorder traversal of $\mathscr{R}_f$. A node is visited before its children, and visiting a node μ now means computing all export lists $L_{\mu\kappa}$. At the time we compute these lists, all import lists are complete and stored with their unprotected edges. In a final walk along the boundary of μ, we project points onto the appropriate unprotected edges, as before.

Some Combinatorial Results. The analysis of the above algorithm requires some topological and combinatorial results about regions. We begin with a combinatorial lemma. Let $e_1, e_2, \ldots, e_k$ be a sequence of k not necessarily distinct symbols. It is a *$DS_2(n)$-sequence* if only n of the symbols e_i are different, $e_i \neq e_{i+1}$ for $1 \leq i \leq k-1$, and there are no four indices $1 \leq i_1 < i_2 < i_3 < i_4 \leq k$ so that $e_{i_1} = e_{i_3} \neq e_{i_2} = e_{i_4}$. The *length* of the sequence is k.

5.1. *The length of any $DS_2(n)$-sequence is at most $2n-1$.*

The upper bound in 5.1 can easily be proved by induction if it is observed that the symbol that is introduced last occurs only once. We use such sequences to bound the number of segments in a region μ.

5.2. *The number of segments of μ is less than $2r(\mu) + |\mu|$.*

Proof. Consider the ordered sequence of segments. Replace each segment by the name of the corresponding unprotected edge. The resulting sequence contains no scattered subsequence of the form

$$\ldots ab \ldots cd \ldots ab \ldots cd \ldots,$$

because otherwise the bounding circles of M_{ab} and M_{cd} would intersect at four or more points. So if we compress repetitions we get a $DS_2(r(\mu))$-sequence. If two consecutive symbols (unprotected edges) are the same, then there must be a vertex of μ separating them. This implies that the total number of segments exceeds the length of the $DS_2(r(\mu))$-sequence by at most $|\mu|$. □

The total number of unprotected edges before the propagation phase is at most $O(mn)$, and it is fairly easy to see that this bound is tight. It is plausible that a single region can have only substantially fewer unprotected edges. We now prove a more general result that implies a single region indeed cannot exceed $O(m+n)$

unprotected edges. For a region μ, define its *excess* $e(\mu) = \max\{0, r(\mu) - 4m\}$. The *total excess* is the sum of the $e(\mu)$ over all regions μ of $\mathscr{H}$. We prove an upper bound on the total excess which is sufficient for our purposes but certainly not tight.

5.3. *The total excess is less than* $36n$.

Proof. To help the discussion we replace each edge of $\mathscr{G}$ by a pair of directed edges, which we call *di-edges*. A di-edge pq *contributes* an edge to a region μ if it contains the edge and along this edge μ lies to the left of pq. Consider the sequence of unprotected edges in μ's boundary, and replace each such edge by the name of the contributing di-edge of $\mathscr{G}$. This results in a sequence with at most $2m$ different symbols. A straightforward topological argument shows that there is no scattered subsequence of the form

$$\ldots pq \ldots st \ldots pq \ldots st \ldots .$$

If we ignore repetitions we have a $DS_2(2m)$-sequence, which implies that the length of the sequence without repetitions is less than $4m$. Anything exceeding this number is counted by $e(\mu)$.

Let ab and cd be two consecutive unprotected edges contributed by the same di-edge, pq. Then:

(i) $b = c$, or
(ii) bc is a protected edge of μ, or
(iii) there are two or more protected edges between b and c.

In case (i) we can charge the projection in stage 1 for the repetition, and there are fewer than $12n$ of them. Each projected point is counted twice, once for each side, so we have fewer than $24n$ repetitions of type (i) in total. In case (iii) we can charge the vertex common to the first two protected edges after b for the repetition. This vertex must be in $V \cup V''$. We have $|V \cup V''| < 4n$, and each such vertex is charged at most twice because it can lie on at most two blocking circles. This implies that there are fewer than $8n$ repetitions of type (iii). In case (ii) bc is a protected edge contributed by pq. We argue in the following that the total number of such edges, over all regions and di-edges, is less than $4n$. This implies the claim.

Since bc is protected, its endpoints lie on a blocking circle C_j bounding D_j. Furthermore, since bc belongs to a region with at least one unprotected edge, it must be one of the edges of the convex hull of the vertices on C_j. Consider D_j and the edges of $\mathscr{H}$ that lie on edges of $\mathscr{G}$ and decompose D_j. Each such edge has both endpoints on C_j. It follows that the dual graph of the decomposition is a free tree. The nodes of the tree are the regions of the decomposition, and the arcs correspond to the edges that decompose D_j. The edge bc corresponds to an arc incident to a leaf of the dual graph. We can bound the number of repetitions of type (ii) by bounding the total number of leaves of the $n - 1$ dual graphs defined for the blocking circles.

To count the total number of leaves, we assume $\mathcal{G}$ is a triangulation. If not, it can be converted into one by adding fewer than $3n - m$ edges; adding these edges can only increase the count. The advantage of a triangulation is that now each interior node of a dual graph has degree 2 or 3. Furthermore, the number of leaves of a dual graph is 2 plus the number of degree-3 nodes. Each degree-3 node of the dual graph for D_j corresponds to a triangle of $\mathcal{G}$, each of whose three sides intersects C_j. A triangle can intersect at most one blocking circle in this manner. Therefore, the total number of degree-3 nodes is at most the number of triangles, i.e., $2n - 5$. This number plus twice the number of blocking circles is less than $4n$, as claimed. □

A region cannot have more than $4n$ vertices shared by adjacent protected edges, because each such vertex is a vertex in $V \cup V''$. Each such vertex is encountered at most twice, which implies $|\mu| \leq 2r(\mu) + 8n$. The result in 5.3 thus implies a bound on the number of edges of a region.

5.4. *For a region μ of $\mathcal{H}$, $|\mu| = O(m + n)$.*

The Final Analysis. Stages 1 and 2 require at most $O(|\mu| r(\mu))$ time per region μ. By definition, $r(\mu) \leq 4m + e(\mu)$, so the time is bounded by a constant times

$$\sum |\mu| m + \sum |\mu| e(\mu).$$

The first sum is $O(m^2 n)$ because $\mathcal{H}$ has only $O(mn)$ edges and stage 1 adds only $O(n)$ to this number. The second sum is $O(n^2)$ because

$$\sum |\mu| e(\mu) = O((n + m) \sum e(\mu))$$

and, by 5.3, $\sum e(\mu) < 36n$. This implies that $O(m^2 n + n^2)$ is an upper bound for the time spent in the first two stages of the algorithm. After that, $O(m^2 n)$ time suffices to compute all import and export lists in stage 3. This implies the main result of this section.

Theorem 5.5. *Let $\mathcal{G} = (V, E)$ be a plane geometric graph with $|V| = n$ and $|E| = m \geq 1$. A point set S of size $O(m^2 n)$ that admits a completion of its Delaunay triangulation conforming to $\mathcal{G}$ can be computed in time $O(m^2 n + n^2)$.*

6. Discussion and Open Problems

The main result of this paper is the existence of $O(m^2 n)$ points that admit a completion of their Delaunay triangulation conforming to a plane geometric graph with n vertices and $m \geq 1$ edges. This result is superficially similar to the triangulation results of [BDE], [BEG], [BE], and [MS]. The best lower bound for the number of points necessary is $\Omega(mn)$, and its proof is fairly straightforward.

It would be interesting to close the gap between the two bounds. The $O(m^2n)$ points can be constructed in time $O(m^2n + n^2)$, provided infinite precision arithmetic in constant time is assumed. This assumption is unrealistic because the number of bits necessary to represent a point accurately increases at each projection along a propagation sequence. Is it possible to construct the points within the same time-bound without this assumption?

A seemingly difficult open problem is the generalization of our polynomial bound to three dimensions. The somewhat easier version of the generalized problem considers a graph whose vertices are embedded as points in $\mathbb{R}^3$, and edges are represented by straight line segments connecting embedded vertices. More relevant, however, is the problem for the crossing-free embedding of a complex consisting of vertices, edges, *and* triangles.

Acknowledgment

We thank Sang-Ho Lee for helpful discussions and Kurt Mehlhorn, Micha Sharir, and Emo Welzl for communicating their improvement to the exponential construction described in Section 3. We are also grateful to two anonymous referees for suggestions on improving the presentation of our results.

References

[BDE] M. Bern, D. Dobkin, and D. Eppstein. Triangulating polygons without large angles. In *Proc. 8th Ann. Symp. on Computational Geometry*, 1992, pp. 222–231.

[BE] M. Bern and D. Eppstein. Polynomial-size nonobtuse triangulation of polygons. In *Proc. 7th Ann. Symp. on Computational Geometry*, 1991, pp. 342–350.

[BEG] M. Bern, D. Eppstein, and J. Gilbert. Provably good mesh generation. In *Proc. 31st Ann. IEEE Symp. on Foundations of Computer Science*, 1990, pp. 231–241.

[BFL] J. D. Boissonnat, O. D. Faugeras, and E. Le Bras-Mehlman. Representing stereo data with the Delaunay triangulation. In *Proc. Internat. Conf. on Robotics and Automation*, 1988, pp. 24–29.

[Bo] J. D. Boissonnat. Shape reconstruction from planar cross sections. *Comput. Vision Graphics Image Process.* **44** (1988), 1–29.

[C] J. Cavendish. Automatic triangulation of arbitrary planar domains for the finite element method. *Internat. J. Numer. Methods Engrg.* **8** (1974), 679–696.

[CLR] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.

[D'AS] E. F. D'Azevedo and R. B. Simpson. On optimal interpolation triangle incidences. *SIAM J. Sci. Statist. Comput.* **10** (1989), 1063–1075.

[De] B. Delaunay. Sur la sphère vide. *Izv. Akad. Nauk SSSR, Otdel. Mat. Est. Nauk* **7** (1934), 793–800.

[Ed] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Heidelberg, 1987.

[ElA] H. ElGindy and D. Avis. A linear algorithm for computing the visibility polygon from a point. *J. Algorithms* **2** (1981), 186–197.

[F] S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica* **2** (1987), 153–174.

[GKS] L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica* **7** (1992), 381–413.

[GS] L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graphics* **4** (1985), 74–123.

[JS] B. Joe and R. B. Simpson. Corrections to Lee's visibility polygon algorithm. *BIT* **27** (1987), 458–473.

[La] C. L. Lawson. Software for C^1 surface interpolation. In *Mathematical Software III*, J. R. Rice, ed. Academic Press, New York, 1977, pp. 161–194.

[Le] D. T. Lee. Visibility of a simple polygon. *Comput. Vision Graphics Image Process.* **22** (1983), 207–221.

[LL] D. T. Lee and A. K. Lin. Generalized Delaunay triangulation for planar graphs. *Discrete Comput. Geom.* **1** (1986), 201–217.

[MS] E. A. Melissaratos and D. L. Souvaine. Coping with inconsistencies: a new approach to produce quality triangulations of polygonal domains with holes. In *Proc. 8th Ann. Symp. on Computational Geometry*, 1992, pp. 202–211.

[NS] L. R. Nackman and V. Srinivasan. Point placement for Delaunay triangulation of polygonal domains. In *Proc. 3rd Canadian Conf. on Computational Geometry*, 1991, pp. 37–40.

[O] A. A. Oloufa. Triangulation applications in volume calculation. *J. Comput. Civil Engrg.* **5** (1991), 103–119.

[PS] F. P. Preparata and M. I. Shamos. *Computational Geometry—an Introduction.* Springer-Verlag, New York, 1985.

[Ra] V. T. Rajan. Optimality of the Delaunay triangulation in $\mathbb{R}^d$. In *Proc. 7th Ann. Symp. on Computational Geometry*, 1991, pp. 357–363.

[Ri] S. Rippa. Minimal roughness property of the Delaunay triangulation. *Comput. Aided Geom. Design* **7** (1990), 489–497.

[Sa] A. Saalfeld. Delaunay edge refinements. In *Proc. 3rd Canadian Conf. on Computational Geometry*, 1991, 33–36.

[SaP] N. Sapidis and R. Perucchio. Delaunay triangulation of arbitrarily shaped planar domains. *Comput. Aided Geom. Design* **8** (1991), 421–437.

[Se] R. Seidel. Constrained Delaunay triangulations and Voronoi diagrams with obstacles. In *1978–1988, 10-Years IIG.* Inst. Inform. Process., Techn. Univ. Graz, 1988, pp. 178–191.

[Si] R. Sibson. Locally equiangular triangulations. *Comput. J.* **21** (1978), 243–245.

[StF] G. Strang and G. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

[V] G. Voronoi. Nouvelles applications des paramètres continus à la thèorie des dormes quadratiques. Deuxième Mémoire: Recherches sur les parallélloèdres primitifs. *J. Reine Angew. Math.* **134** (1908), 198–287.

Received May 10, 1992, and in revised form November 13, 1992.