数理解析研究所講究録
1120 巻 1999 年 182-190

一様な三角形メッシュ生成アルゴリズム

Approximating Uniform Triangular Meshes in Polygons

Franz AURENHAMMER[1] 加藤直樹 [2] Naoki KATOH 大崎純 [2] Makoto OHSAKI 徐寅峰 [3] Yin-Feng XU

[1]Graz Univ. of Technology
Klosterwiesgasse 32/2, A-8010 Graz, Austria
auren@igi.tu-graz.ac.at

[2] 京都大学大学院工学研究科
〒 606-8501 京都市左京区吉田本町
{naoki,ohsaki}@archi.kyoto-u.ac.jp

[3] 西安交通大学管理学院
Xi'an, 710049 P.R.China
yfxu@xjtu.edu.cn

摘要: Given a convex polygon $\boldsymbol{P}$ in the plane and an integer n, we consider the problem of triangulating $\boldsymbol{P}$ using n Steiner points under the following optimality criteria: (1) minimizing the ratio of the maximum edge length to the minimum one, (2) minimizing the maximum edge length, and (3) minimizing the maximum triangle perimeter. We establish a relation of these problems to a certain extreme packing problem for $\boldsymbol{P}$. Based on this relationship, we develope a heuristic producing constant approximations for any of the optimality criteria above (provided n is chosen sufficiently large). That is, the triangular mesh produced is uniform in these respects. The method is easy to implement and runs in $O(n^2 \log n)$ time and $O(n)$ space. The observed runtime is much less. Moreover, for criterion (1) the method works – within the same complexity and approximation bounds – for arbitrary polygons with possible holes, and for criteria (2) and (3) it does so for a large subclass.

キーワード: triangulation, meshgeneration, Voronoi diagram, approximation algorithm

1 Introduction

Given a convex polygon $\boldsymbol{P}$ in the plane and a positive integer n, we consider the problem of generating a triangular mesh for the interior of $\boldsymbol{P}$ using n Steiner points such that certain optimality criteria concerning uniformity of edge lengths are satisfied. In other words, under certain optimality criteria, we want to find a set S_n of n points inside $\boldsymbol{P}$ as well as a triangulation of $\boldsymbol{P}$ using S_n. The problems we consider are formalized as follows: Let V be the set of vertices of $\boldsymbol{P}$, and let $\mathcal{T}$ denote the set of all possible triangulations of $S_n \cup V$. When a point set S_n inside $\boldsymbol{P}$ is fixed, we suppose that we want to minimize the respective objective function over all triangulations in $\mathcal{T}$. We shall consider the following three objective functions: (1) ratio of the maximum edge length to the minimum one, (2) maximum edge length, and (3) maximum triangle perimeter. Let $l(e)$ denote the (Euclidean) length of edge e, and let $peri(\Delta)$ be the perimeter of triangle Δ. The problems under these three criteria read as follows.

Problem 1:

$$\min_{S_n \subset \boldsymbol{P}} \min_{T \in \mathcal{T}} \max_{e,f \in T} \frac{l(e)}{l(f)}.$$

Problem 2:

$$\min_{S_n \subset \boldsymbol{P}} \min_{T \in \mathcal{T}} \max_{e \in T} l(e).$$

Problem 3:

$$\min_{S_n \subset \boldsymbol{P}} \min_{T \in \mathcal{T}} \max_{\Delta \in T} peri(\Delta).$$

We will first develope a heuristic called *canonical Voronoi insertion* which approximately solves a certain extreme packing problem for point sets within $\boldsymbol{P}$. The method is similar to the one used in Gonzalez [10] and Feder and Greene [8] developed for clustering problems. We then show how to modify the heuristic, to produce a set of n points whose Delaunay triangulation within $\boldsymbol{P}$ constitutes a constant approximation for any of the three problems stated above. Respective approximation factors of 6, $4\sqrt{3}$, and $6\sqrt{3}$ are proven, provided n is sufficiently large. As a byproduct, the solution we construct is a triangulation of constant vertex degree. With minor modifications, our method works for arbitrary polygons (with possible holes), and yields the same approximation result for Problem 1. Concerning Problems 2 and 3, the approximation factors above can be guaranteed for a restricted class of non-convex polygons.

Generating triangulations is one of fundamental problems in computational geometry, and has been extensively studied; see e.g. the survey article by Bern and Eppstein [3]. Main fields of applications are finite element methods and computer aided design. In finite element methods, for example, it is desirable to generate triangulations that do not have too large or too small angles. Along this direction, various algorithms have been reported [4, 13, 6, 2, 5, 16]. Restricting angles means bounding the edge length ratio for the individual triangles, but not necessarily for a triangulation in global, which might be desirable in some applications. That is, the produced triangulation need not be uniform concerning the edge ratio or the perimeter ratio of its triangles. Chew [6] and Melisseratos and Souvaine [13] construct uniform triangular meshes in the weaker sense that only upper bounds on the triangle size are required.

A particular application of uniform triangulation arises in designing structures such as plane trusses with triangular units, where it is required to determine the shape from aesthetic points of view under the constraints concerning stress and nodal displacement. The plane truss can be viewed as a triangulation of points in the plane by regarding truss members and nodes as edges and points, respectively. When focusing on the shape, edge lengths should be as equal as possible from the viewpoint of design, mechanics and manufacturing; see [14, 15]. In such applications, the locations of the points are usually not fixed, but can be viewed as decision variables. In view of this application field, it is quite natural to consider Problems 1, 2, and 3. To the knowledge of the authors, the problems dealt with in this paper have not been studied in the field of computational geometry. The mesh refinement algorithms in Chew [6] and in Ruppert [16] are similar in spirit to our Voronoi insertion method, but aim at different optimality criteria. A general advantage of the meshes generated by their methods as well as ours is the absence of favoured edge orientations. This advantage is not shared by grid-based or quadtree-based methods which are frequently used.

Finding an optimal solution for any of the three problems seems to be difficult in view of the NP-completeness of packing problems in the plane; see e.g. Johnson [12]. For the case of a fixed point set, minimizing the maximum edge length is known to be solvable in quadratic time; see Edelsbrunner and Tan [7]. Nooshin et al. [14] developed a potential-based heuristic method for Problem 2, but did not give a theoretical guarantee for the obtained solution.

The following notation will be used throughout. For two points x and y in the plane, let $l(x, y)$ denote their Euclidean distance. The minimum (non-zero) distance between two point sets X and Y is defined as $l(X, Y) = \min\{l(x, y) \mid x \in X, y \in Y, x \neq y\}$. When X is a singleton set $\{x\}$ we simply write $l(X, Y)$ as $l(x, Y)$. Note that $l(X, X)$ defines the minimum interpoint distance among the point set X.

2 Canonical Voronoi Insertion and Extreme Packing

In this section, we consider the following *extreme packing problem*. Let $\boldsymbol{P}$ be a (closed) convex polygon with vertex set V.

$$\text{Maximize } l(V \cup S_n, V \cup S_n)$$
$$\text{subject to a set } S_n \text{ of } n \text{ points within } \boldsymbol{P}.$$

In other words, the problem asks for a packing of n circles with centers in $\boldsymbol{P}$ such that the smallest radius is maximum. We shall give a 2-approximation algorithm for this problem using *canonical Voronoi insertion*. In Section 3 we then show that the point set S_n produced by this algorithm, as well as the Delaunay triangulation induced by S_n within $\boldsymbol{P}$, can be modified to give an approximate solution for the three problems addressed in Section 1.

The algorithm determines the location of the point set S_n in a greedy manner. Namely, starting with an empty set S, it repeatedly places a new point inside $\boldsymbol{P}$ at the position which is farthest from the set $V \cup S$. The idea of the algorithm originates with Gonzalez [10] and Feder and Greene [8], and was developed for approximating minimax k-clusterings. Comparable insertion strategies are also used for mesh generation in Chew [6] and in Ruppert [16], there called *Delaunay refinement*. Their strategies aim at different quality measures, however, and insertion does not take place in a canonical manner.

For approximation results concerning packings where the size of the objects rather than their number is prescribed see e.g. Hochbaum and Maass [11]. Various results on the size of circle packings are summarized in Fejes Tóth [9].

The algorithm is formally described below. It uses the Voronoi diagram of the current point set to select the next point to be inserted. We assume familiarity with the basic properties of a Voronoi diagram and its dual, the Delaunay triangulation, and refer to the survey paper [1].

Algorithm INSERT

Step 1: Initialize $S := \emptyset$.

Step 2: Compute the Voronoi diagram $\mathrm{Vor}(V \cup S)$ of $V \cup S$.

Step 3: Find the set B of intersection points between edges of $\mathrm{Vor}(V \cup S)$ and the boundary of $\boldsymbol{P}$. Among the points in B and the vertices of $\mathrm{Vor}(V \cup S)$ inside $\boldsymbol{P}$, choose the point u which maximizes $l(u, V \cup S)$.

Step 4: Put $S := S \cup \{u\}$ and return to Step 2 if $|S| < n$.

Let p_j and S_j, respectively, denote the point chosen in Step 3 and the set obtained in Step 4 at the j-th iteration of the algorithm. For an arbitrary point $x \in \boldsymbol{P}$ define the *weight* of x with respect to S_j as $w_j(x) = l(x, S_j \cup V)$. That is, $w_j(x)$ is the radius of the largest circle centered at x which does not enclose any point from $S_j \cup V$. By definition of a Voronoi diagram, the point p_j maximizes $w_{j-1}(x)$ over all $x \in \boldsymbol{P}$. Let

$$d_n = l(S_n \cup V, S_n \cup V) \tag{1}$$

be the minimum interpoint distance realized by $S_n \cup V$. Furthermore, denote by S_n^* the optimal solution for the extreme packing problem for $\boldsymbol{P}$ and let d_n^* denote the corresponding objective value. The following approximation result might be of interest in its own right. Its proof is an adaptation of techniques in [10, 8] and contains observations that will be used in our further analysis.

Theorem 1 *The solution S_n obtained by Algorithm INSERT is a 2-approximation of the extreme packing problem for $\boldsymbol{P}$. That is, $d_n \geq d_n^*/2$.*

Proof. We claim that p_n realizes the minimum (non-zero) distance from S_n to $S_n \cup V$. Equivalently, the claim is

$$w_{n-1}(p_n) = l(S_n, S_n \cup V). \tag{2}$$

To see this, assume that the minimum distance is realized by points p_k and p_j different from p_n. Without loss of generality, let p_k be inserted after p_j by the algorithm. Then we get $w_{k-1}(p_k) \leq l(p_k, p_j) < l(p_n, S_{n-1} \cup V) = w_{n-1}(p_n)$. On the other hand, the sequence of weights chosen by the algorithm must be non-increasing. More exactly, $w_{k-1}(p_k) \geq w_{k-1}(p_n) \geq w_{n-1}(p_n)$. This is a contradiction.

From the trivial observations $d_n = \min\{l(S_n, S_n \cup V), l(V, V)\}$ and $l(V, V) \geq d_n^* \geq d_n$ we now get $d_n = \min\{w_{n-1}(p_n), d_n^*\}$ by (2). As p_n maximizes $w_{n-1}(x)$ for all points $x \in \boldsymbol{P}$, the lemma below completes the proof of the theorem. ◻

Lemma 1 *For any set $S \subset \boldsymbol{P}$ of $n-1$ points there exists a point $x \in \boldsymbol{P}$ with $l(x, S \cup V) \geq d_n^*/2$.*

Proof. Suppose that the lemma is not true. Then the point $x \in \boldsymbol{P}$ farthest from S satisfies

$$l(x, S \cup V) < d_n^*/2. \tag{3}$$

Let r be the value of the left-hand side of (3). For each point $p \in S \cup V$ draw a circle centered at p

with radius $r + \epsilon$, where ϵ is a sufficiently small positive number that satisfies $r + \epsilon < d_n^*/2$. The union of these circles covers the whole area of $\boldsymbol{P}$ as each uncovered point would be farther from $S \cup V$ than is x. On the other hand, one of these circles must cover two points of $S_n^* \cup V$, as the number of points in this set is by one larger than the number of circles. The distance between these two points is at most $2(r + \epsilon)$, which is less than d_n^* by (3). This contradicts the definition of d_n^*. ◻

3 Delaunay Triangulation of Bounded Edge Ratio

Our aim is to show that Algorithm INSERT is capable of producing a point set appropriate for Problems 1, 2, and 3. To this end, we first investigate the Delaunay triangulation $\mathrm{DT}(S_n \cup V)$ of $S_n \cup V$. This triangulation is implicitly constructed by the algorithm, as being the dual structure of $\mathrm{Vor}(S_n \cup V)$. However, $\mathrm{DT}(S_n \cup V)$ need nor exhibit good edge length properties. We therefore prescribe the placement of the first k inserted points, and show that Algorithm INSERT completes them to a set of n points whose Delaunay triangulation has its edge lengths controlled by the minimum interpoint distance d_n for $S_n \cup V$.

For $1 \leq j \leq n$, consider the triangulation $\mathrm{DT}(S_j \cup V)$. Let us classify a triangles Δ of $\mathrm{DT}(S_j \cup V)$ as either *critical* or *non-critical*, depending on whether the Voronoi vertex dual to Δ (i.e., the circumcenter of Δ) lies outside of the polygon $\boldsymbol{P}$ or not. Whereas edges of critical triangles can be arbitrarily long, edge lengths are bounded in non-critical triangles.

Lemma 2 *No edge e of a non-critical triangle Δ of $DT(S_j \cup V)$ is longer than $2 \cdot w_{j-1}(p_j)$.*

PROOF. Let $e = (p, q)$ and denote with x the Voronoi vertex dual to Δ. As x lies inside of $\boldsymbol{P}$, we get $l(x,p) = l(x,q) = w_{j-1}(x) \leq w_{j-1}(p_j)$, by the choice of point p_j in Step 3 of Algorithm INSERT. The triangle inequality now implies $l(p,q) \leq 2 \cdot w_{j-1}(p_j)$. ◻

We make an observation on critical triangles. Consider some edge e of $\mathrm{DT}(S_j \cup V)$ on the boundary of $\boldsymbol{P}$. Edge e cuts off some part of the diagram $\mathrm{Vor}(S_j \cup V)$ that is outside of $\boldsymbol{P}$. If that part contains Voronoi vertices then we define the *critical region*, $R(e)$, for e as the union of all the (critical) triangles that are dual to these vertices. Notice that each critical triangle of $\mathrm{DT}(S_j \cup V)$ belongs to a unique critical region.

Lemma 3 *No edge f of a critical triangle in $R(e)$ is longer than $l(e)$.*

PROOF. Let p be an endpoint of f. Then the region of p in $\mathrm{Vor}(S_j \cup V)$ intersects e. Let x be a point in this region but outside of $\boldsymbol{P}$. There is a circle around x that encloses p but does not enclose any endpoint of e. Within $\boldsymbol{P}$, this circle is completely covered by the circle C with diameter e. This implies that p lies in C. As the distance between any two points in C is at most $l(e)$, we get $l(f) \leq l(e)$. ◻

Let us further distinguish between *interior* triangles and *non-interior* ones, the former type having no two endpoints on the boundary of $\boldsymbol{P}$. The shortest edge of an interior triangle can be bounded as follows.

Lemma 4 *Each edge e of an interior triangle Δ of $DT(S_j \cup V)$ has a length of at least $w_{j-1}(p_j)$.*

PROOF. We have $l(e) \geq l(S_j, S_j \cup V)$, because Δ has no two endpoints on $\boldsymbol{P}$'s boundary. But from (2) we know $l(S_j, S_j \cup V) = w_{j-1}(p_j)$. ◻

We are now ready to show how a triangulation with edge lengths related to d_n can be computed. First, Algorithm INSERT is run on $\boldsymbol{P}$, in order to compute the value d_n. We assume than n is chosen sufficently large to assure $d_n \leq l(V,V)/2$. This assumption is not unnatural as the shortest edge of the desired triangulation cannot be longer than the shortest edge of $\boldsymbol{P}$. After having d_n available, k points $p'_1, \ldots, p'_k$ are placed on the boundary of $\boldsymbol{P}$, with consecutive distances between $2 \cdot d_n$ and $3 \cdot d_n$, and such that $l(V', V') \geq d_n$ holds, for $V' = V \cup \{p'_1, \ldots, p'_k\}$. Notice that such a placement is always possible. Finally, $n - k$ additional points $p'_{k+1}, \ldots, p'_n$ are produced by rerunning Algorithm INSERT after this placement.

For $1 \leq j \leq n$, let $S'_j = \{p'_1, \ldots, p'_j\}$. Define $w(x) = l(x, S'_n \cup V)$ for a point $x \in \boldsymbol{P}$. The value of $w(p'_n)$ will turn out to be crucial for analyzing the edge length behavior of the triangulation

$DT(S'_n \cup V)$. The lemma below asserts that $w(p'_n)$ is small if n exceeds twice the number k of prescribed points.

Lemma 5 *Suppose $n \geq 2k$. Then $w(p'_n) \leq 3 \cdot d_n$.*

PROOF. The point set S_n produced by Algorithm INSERT in the first run is large enough to ensure $d_n < l(V, V)$. So we get $d_n = w_{n-1}(p_n)$ from (2). As point p_n maximizes $w_{n-1}(x)$ for all $x \in \boldsymbol{P}$, the $n + |V|$ circles centered at the points in $S_n \cup V$ and with radii d_n completely cover the polygon $\boldsymbol{P}$. Let $d_n = 1$ for the moment. Then

$$A(\boldsymbol{P}) \leq \pi(n + |V|) - A' \qquad (4)$$

where $A(\boldsymbol{P})$ is the area of $\boldsymbol{P}$, and A' denotes the area outside of $\boldsymbol{P}$ which is covered by the circles centered at V.

Assume now $w(p'_n) > 3 \cdot d_n$. Draw a circle with radius $\frac{3}{2}d_n$ around each point in $S'_n \setminus S'_k$. Since $w(p'_n) = l(S'_n \setminus S'_k, S'_n \cup V)$ by (2), these circles are pairwise disjoint. By the same reason, and because boundary distances defined by $V' = V \cup S'_k$ are at most $3 \cdot d_n$, these circles all lie completely inside $\boldsymbol{P}$. Obviously, these circles are also disjoint from the $|V|$ circles of radius d_n centered at V. Finally, the latter circles are pairwise disjoint, since $d_n \leq l(V, V)/2$. Consequently,

$$A(\boldsymbol{P}) \geq \frac{9}{4}\pi(n - k) + A'' \qquad (5)$$

where A'' denotes the area inside of $\boldsymbol{P}$ which is covered by the circles centered at V. Combining (4) and (5), and observing $A' + A'' = \pi \cdot |V|$ now implies $n < 2k$, a contradiction. □

It has to be observed that the number k depends on n. The following fact guarantees the assumption in Lemma 5, provided n is sufficently large. Let $B(\boldsymbol{P})$ denote the perimeter of $\boldsymbol{P}$.

Lemma 6 *The condition $d_n \leq A(\boldsymbol{P})/(\pi \cdot B(\boldsymbol{P}))$ implies $n \geq 2k$.*

PROOF. From (4) we obtain

$$n \geq \frac{A(\boldsymbol{P})}{\pi \cdot (d_n)^2} - |V|.$$

To get a bound on k, observe that at most $l(e)/2d_n - 1$ points are placed on each edge e of $\boldsymbol{P}$. This sums up to

$$k \leq \frac{B(\boldsymbol{P})}{2d_n} - |V|.$$

Simple calculations now show that the condition on d_n stated in the lemma implies $n \geq 2k$. □

Theorem 2 *Suppose n is large enough to assure the conditions $d_n \leq l(V, V)/2$ and $d_n \leq A(\boldsymbol{P})/(\pi \cdot B(\boldsymbol{P}))$. Then no edge in the triangulation $T^+ = DT(S'_n \cup V)$ is longer than $6 \cdot d_n$. Moreover, T^+ exhibits an edge length ratio of 6.*

PROOF. Two cases are distinguished, according to the value of $w(p'_n)$.

Case 1: $w(p'_n) < d_n$. Concerning upper bounds, Lemma 2 implies $l(e) \leq 2 \cdot w(p'_n) < 2 \cdot d_n$ for all edges e belonging to non-critical triangles of T^+. If e belongs to some critical triangle, Lemma 3 shows that $l(e)$ cannot be larger than the maximum edge length on the boundary of $\boldsymbol{P}$, which is at most $3 \cdot d_n$ by construction. Concerning lower bounds, Lemma 4 gives $l(e) \geq w(p'_n)$ for edges of interior triangles. We know $w(p'_n) \geq d^*_n/2$ from Lemma 1, which implies $l(e) \geq d_n/2$ because $d^*_n \geq d_n$. For edges spanned by V', we trivially obtain $l(e) \geq d_n$ as $l(V', V') \geq d_n$ by construction.

Case 2: $w(p'_n) \geq d_n$. The upper bound $2 \cdot w(p'_n)$ for non-critical triangles now gives $l(e) \leq 6 \cdot d_n$, due to Lemmas 5 and 6. The lower bound for interior triangles becomes $l(e) \geq w(p'_n) \geq d_n$. The remaining two bounds are the same as in the former case. □

The time complexity of computing the triangulation T^+ is dominated by the runtime of Algorithm INSERT. Let us see how fast this algorithm can be implemented.

It is sufficient to consider Steps 2 and 3. In the very first iteration of the algorithm, both steps can be accomplished in $O(|V| \log |V|)$ time. In each further iteration j we update the current Voronoi diagram under the insertion of a new point p_j in Step 2, as well as a set of weights for the Voronoi vertices and relevant polygon boundary points in Step 3.

Consider Step 2. Since we already know the location of the new point p_j in the current Voronoi diagram, the region of p_j in the updated diagram can be integrated in time proportional to the number of edges of this region. This number is the degree of p_j in the resulting Delaunay triangulation, $\deg(p_j)$.

In Step 3 we need to assign the current weight $w(u)$ to each new Voronoi vertex or boundary intersection point u. Clearly $w(u)$ can be determined in constant time by calculating the radius of the corresponding empty circle. The current set of weights is organized in some priority queue. When processing the point p_j we need to insert and delete $O(\deg(p_j))$ weights, and then select the largest one in the next iteration. This gives a runtime of $O(\deg(p_j) \cdot \log(j + |V|))$ for updating the weights, and thus dominates Step 2.

The following lemma bounds the number of constructed triangles, of a certain type. Let us call a triangle *good* if it is both interior and non-critical.

Lemma 7 *The insertion of each point p_j creates only a constant number of good triangles.*

PROOF. Consider the endpoints of all good triangles incident to p_j in $\mathrm{DT}(S_j \cup V)$, and let X be the set of all such endpoints interior to $\boldsymbol{P}$. Then $l(X, X) \geq l(S_j, S_j) \geq w_{j-1}(p_j)$, due to (2). On the other hand, by Lemma 2, X lies in the circle of radius $2 \cdot w_{j-1}(p_j)$ around p_j. As a consequence, $|X|$ is constant. The number of good triangles incident to p_j is at most $2 \cdot |X|$, as one such triangle would have two endpoints on $\boldsymbol{P}$'s boundary, otherwise. □

For most choices of $\boldsymbol{P}$ and n, the good triangle type will be most frequent. This is supported by the following fact.

Lemma 8 *Let Δ be a critical triangle of $DT(S_j \cup V)$, and let q be any endpoint of Δ. The normal distance of q from the boundary of $\boldsymbol{P}$ is at most $w_{j-1}(p_j)$.*

PROOF. As Δ is critical, there is an edge of the region of q in $\mathrm{Vor}(S_j \cup V)$ which intersects the boundary of $\boldsymbol{P}$. Consider such an intersection point x. We have $l(q, x) = w_{j-1}(x) \leq w_{j-1}(p_j)$, from the way p_j is selected by Algorithm INSERT. On the other hand, the normal distance of q from the boundary of $\boldsymbol{P}$ cannot be larger than $l(q, x)$. □

The number of critical or non-interior triangles incident to p_j in $\mathrm{DT}(S_j \cup V)$ might be high, however. Still, the degree of each point in the final triangulation T^+ is constant, as the longest edge in T^+ is bounded by a constant multiple of the respective minimum interpoint distance (which equals the shortest edge length in T^+ because T^+ is Delaunay).

In conclusion, we obtain a runtime bound of $O(n^2 \log n)$ and a space complexity of $O(n)$. However, Lemmas 7 and 8 suggest a runtime of $O(\log n)$ in most iterations.

Concerning the choice of n, Theorem 2 may hold for much smaller values of n than is required by the sufficient condition $d_n \leq l(V, V)/2$ and $d_n \leq A(\boldsymbol{P})/(\pi \cdot B(\boldsymbol{P}))$. In a particular application, this can be tested efficiently, by repeatedly doubling the chosen value of n and each time examining the edge lengths in T^+.

4 Approximation Results

Let us now return to the three optimization problems for the polygon $\boldsymbol{P}$ posed in the introduction. We will rely on Theorem 2 in the following. Recall that, in order to make the theorem hold, we have to choose n sufficiently large.

Theorem 3 *The triangulation T^+ approximates the optimal solution for Problem 1 by a factor of 6.*

PROOF. Theorem 2 guarantees for T^+ an edge length ratio of 6, and for no triangulation this ratio can be smaller than 1. □

We now turn our attention to Problem 2. Let the point set $\tilde{S}$ in conjunction with the triangulation $\tilde{T}$ of $\tilde{S} \cup V$ be the corresponding optimum solution. Let d_{long} denote the optimum objective value, that is, d_{long} measures the longest edge in $\tilde{T}$. The lemma below relates d_{long} to the optimum value d_n^* for the extreme packing problem for $\boldsymbol{P}$.

Lemma 9

$$d_{long} \geq \frac{\sqrt{3}}{2} d_n^*.$$

PROOF. Suppose the lemma is not true. Let $r = \frac{1}{\sqrt{3}} d_{long}$. For each point $p \in \tilde{S} \cup V$ draw a circle with radius r around p. Let $\tilde{C}$ denote the set of these circles. For each triangle Δ of $\tilde{T}$ its area is entirely covered by the circles of $\tilde{C}$ centered at its three endpoints. This is because the maximum distance from a point within Δ to its endpoints is at most $1/\sqrt{3}$ times the length of its longest edge. So $\tilde{\mathcal{C}}$ entirely covers the area of $\boldsymbol{P}$.

Next, consider the optimal solution S_n^* for the extreme packing problem. Again, around each point in $S_n^* \cup V$ draw a circle with radius r. Let $\mathcal{C}^*$ be the resulting set of circles. Circles in $\mathcal{C}^*$ neither overlap nor touch each other since $r < d_n^*/2$ holds by our assumption that the lemma is false. So $\mathcal{C}^*$ does not entirely cover the area of $\boldsymbol{P}$.

Let $\boldsymbol{Q}$ be the convex hull of $\tilde{\mathcal{C}}$ (and thus of $\mathcal{C}^*$). We now consider what happens in the region $\boldsymbol{Q} \setminus \boldsymbol{P}$. Let e be an arbitrary edge of $\boldsymbol{P}$, and let $\boldsymbol{R}$ denote the rectangle spanned by e and the boundary edge of $\boldsymbol{Q}$ parallel to e. Since $\boldsymbol{P}$ is convex, these rectangles are mutually disjoint. For edge e we have $\frac{2}{\sqrt{3}} d_{long} < d_n^* \leq l(e)$. So there must exist points from $\tilde{S}$ on e such that the distance between consecutive points is at most d_{long}. Their number is at least $\lceil l(e)/d_{long} \rceil - 1$. Consequently, the number of circles of $\tilde{\mathcal{C}}$ whose centers are on e is at least $\lceil l(e)/d_{long} \rceil + 1$. As these circles overlap if their centers are neighbored on e, the area of $\boldsymbol{R}$ covered by circles of $\tilde{\mathcal{C}}$ satisfies

$$\tilde{R} > \frac{\pi \cdot (d_{long})^2}{6} \cdot \lceil l(e)/d_{long} \rceil. \tag{6}$$

On the other hand, we claim that the number of circles of $\mathcal{C}^*$ that intersect e is at most $\lceil l(e)/d_{long} \rceil + 1$. Let $q_1, \ldots, q_h$ be the vertical projections of their centers $p_1, \ldots, p_h$ onto e (in consecutive order). Consider the two circles $C_i, C_{i+1} \in \mathcal{C}^*$ around p_i and p_{i+1}, respectively; see Figure 1.

Since C_i and C_{i+1} are disjoint, we have

$$l(p_i, p_{i+1}) > \frac{2}{\sqrt{3}} d_{long}. \tag{7}$$

Both C_i and C_{i+1} intersect e hence

$$|l(p_i, q_i) - l(p_{i+1}, q_{i+1})| \leq \frac{1}{\sqrt{3}} d_{long}. \tag{8}$$

With (7) and (8), the Pythagorean theorem implies

$$l(q_i, q_{i+1}) > d_{long}.$$

Therefore the claimed upper bound on the number of circles that intersect e follows. Since the circles in $\mathcal{C}^*$ are pairwise disjoint, the area in $\boldsymbol{R}$ covered by $\mathcal{C}^*$ satisfies

$$R^* < \frac{\pi \cdot (d_{long})^2}{6} \cdot \lceil l(e)/d_{long} \rceil. \tag{9}$$

Thus, from (6) and (9), $R^* < \tilde{R}$ follows.

We conclude that the total area covered by $\mathcal{C}^*$ is less than the total area covered by $\tilde{\mathcal{C}}$. But this is a contradiction because the cardinalities of these sets are the same, and circles in $\tilde{\mathcal{C}}$ overlap whereas circles in $\mathcal{C}^*$ do not. ◻

We strongly conjecture that the statement of Lemma 9 can be strengthened to $d_{long} \geq d_n^*$, which will improve the approximation ratio in Theorems 4 and 5 below.

Theorem 4 *The triangulation T^+ constitutes a $4\sqrt{3}$ approximation for Problem 2.*

PROOF. Let e_{max} denote the longest edge in T^+. By Theorem 2 we have $l(e_{max}) \leq 6 \cdot d_n$. Trivially $d_n \leq d_n^*$ holds, and Lemma 9 implies the theorem, $l(e_{\max})/d_{long} \leq 4\sqrt{3}$. ◻

Finally let us consider Problem 3. Let d_{peri} denote the optimum objective value for this problem. We show the following:

Theorem 5 *The triangulation T^+ gives a $6\sqrt{3}$-approximation for Problem 3.*

PROOF. For any triangulation of $\boldsymbol{P}$ with n Steiner points, its longest edge cannot be shorter than $\frac{\sqrt{3}}{2} \cdot d_n^*$ by Lemma 9. This implies $d_{peri} \geq \sqrt{3} \cdot d_n^*$ by the triangle inequality. On the other hand, for the longest edge e_{max} of T^+ we have $l(e_{max}) \leq 6 \cdot d_n^*$ due to Theorem 2. The longest triangle perimeter δ_{max} that occurs in T^+ is at most $3 \cdot l(e_{max})$. In summary, $\delta_{max}/d_{peri} \leq 6 \cdot \sqrt{3}$. ◻

We conclude this section by mentioning an approximation result concerning minimum-weight triangulations.

Theorem 6 *Let S^+ be the vertex set of T^+ and let $MWT(S^+)$ denote the minimum-weight triangulation of S^+. Then T^+ is a 6-length approximation for $MWT(S^+)$.*

PROOF. Let e_{min} be the shortest edge in T^+. Then $l(e_{min})$ is the minimum interpoint distance in S^+, because T^+ is Delaunay. So any edge e of $\mathrm{MWT}(S^+)$ satisfies $l(e) \geq l(e_{min})$. On the other hand, any edge e' of T^+ fulfills $l(e') \leq 6 \cdot l(e_{min})$, by Theorem 2. It remains to be observed that every triangulation of S^+ realizes the same number of edges. ◻

5 Discussion and Extensions

We have considered the problem of generating length-uniform triangular meshes for the interior of convex polygons. A unifying algorithm capable of computing constant approximations for these problems has been developed. The basic idea is to relate the length of triangulation edges to the optimum extreme packing distance. The proposed heuristic is easy to implement and seems to produce acceptably good triangular meshes as far as computational experiments are concerned.

In practical applications, more general input polygons need to be triangulated. In fact, our algorithm works with minor modification for arbitrary polygons with possible holes. Convexity is used solely in the proof of Lemma 9. As a consequence, Theorems 1 and 2, the approximation result for Problem 1, and Theorem 6 still hold. The modification needed is that *visible distances* in a non-convex polygon $\boldsymbol{P}$ should be considered only, in the proofs as well as concerning the algorithm. That is, for the point sets $S \subset \boldsymbol{P}$ in question, the Delaunay triangulation of $S \cup V$ *constrained by* $\boldsymbol{P}$ has to be utilized rather than $\mathrm{DT}(S \cup V)$.

The proof of Lemma 9 (and with it the approximation results for Problems 2 and 3) still go through for non-convex polygons $\boldsymbol{P}$ with interior angles of at most $\frac{3\pi}{2}$, provided n is large enough to make the value $\frac{2}{\sqrt{3}}d_{long}$ fall short of the minimum distance between non-adjacent edges of $\boldsymbol{P}$. The bottleneck is inequality (9) which need not hold if the rectangular regions around $\boldsymbol{P}$ overlap. We pose the question of establishing a version of Lemma 9 for general non-convex polygons, and of improving the respective bound $\frac{\sqrt{3}}{2}$ for the convex case.

From the viewpoint of applications to design of structures, it is also important to generate a triangular mesh for approximating surfaces such as large-span structures. This topic is left to further research.

Acknowledgements

This research was partially supported by the Austrian Spezialforschungsbereich F003, Optimierung und Kontrolle, and the Grant-in-Aid for Scientific Research on Priority Areas (B) (No. 10205214) by the Ministry of Education, Science, Sports and Culture of Japan. In addition, the fourth author was also supported by NSF of China (No. 19731001) and the Japan Society for the Promotion of Science of Japan. We greatfully acknowledge all these support.

References

[1] F. Aurenhammer, "Voronoi diagrams – a survey of a fundamental geometric data structure", *ACM Computing Surveys* 23 (1991), 345-405.

[2] M. Bern, D. Dobkin and D. Eppstein, "Triangulating polygons without large angles", *Intl. J. Comput. Geom. and Appl.* 5 (1995), 171-192.

[3] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation", in D.-Z. Du (ed.), *Computing in Euclidean Geometry*, World Scientific Publishing, 1992, 47-123.

[4] M. Bern, D. Eppstein and J.R. Gilbert, "Provably good mesh generation", *Journal of Computer and System Sciences* 48 (1994), 384-409.

[5] M. Bern, S. Mitchell and J. Ruppert, "Linear-size nonobtuse triangulation of polygons", Proceedings of the 10th Ann. ACM Symposium on Computational Geometry (1994), 221-230.

[6] P. Chew, "Guaranteed-Quality Mesh Generation for Curved Surfaces", Proceedings of the 9th Ann. ACM Symposium on Computational Geometry (1993), 274-280.

[7] H. Edelsbrunner and T.S. Tan, "A quadratic time algorithm for the minmax length triangulation", *SIAM Journal on Computing* 22 (1993), 527-551.

[8] T. Feder and D.H. Greene, "Optimal Algorithms for Approximate Clustering", Proceedings of the 20th Ann. ACM Symposium STOC (1988), 434-444.

[9] G. Fejes Tóth, "Packing and covering", in *Handbook of Discrete and Computational Geometry*, J.E. Goodman and J. O'Rourke (eds.), CRC Press Series on Discrete Mathematics and its Applications, 1997, 19-41.

[10] T. Gonzalez, "Clustering to minimize the maximum intercluster distance", *Theoretical Computer Science* 38 (1985), 293-306.

[11] D.S. Hochbaum and W. Maass, "Approximation schemes for covering and packing problems in image processing and VLSI". *Journal of the ACM* 32 (1985), 130-136.

[12] D.S. Johnson, "The NP-completeness column: An ongoing guide", *Journal of Algorithms* 3 (1982), 182-195.

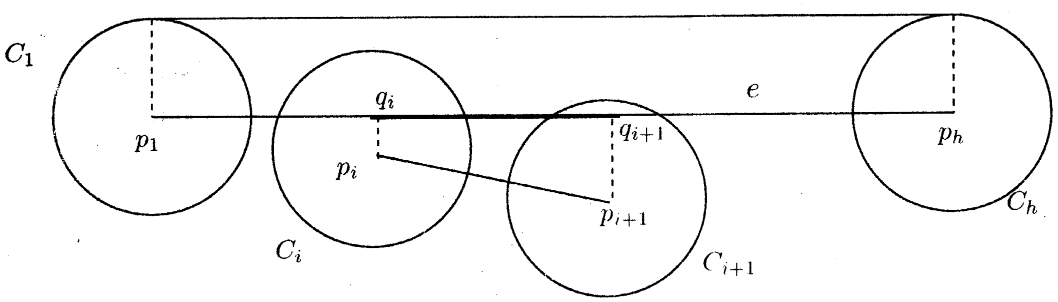

Figure 1: Illustration of circles C_i and C_{i+1}.

[13] E. Melisseratos and D. Souvaine, "Coping with inconsistencies: A new approach to produce quality triangulations of polygonal domains with holes", Proceedings of the 8th Ann. ACM Symposium on Computational Geometry (1992),202-211.

[14] H. Nooshin, K. Ishikawa, P.L. Disney and J.W. Butterworth, "The traviation process", *Journal of the International Association for Shell and Spatial Structures* 38 (1997), 165-175.

[15] M. Ohsaki, T. Nakamura and M. Kohiyama, "Shape optimization of a double-layer space truss described by a parametric surface", *International Journal of Space Structures* 12 (1997), 109-119.

[16] J. Ruppert, "A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation", *Journal of Algorithms* 18 (1995), 548-585.