Approximating Uniform Triangular Meshes in Polygons

Franz Aurenhammer[1], Naoki Katoh[2] and Hiromichi Kojima[2], Makoto Ohsaki[2], and Yinfeng Xu[3]

[1] Institute for Theoretical Computer Science, Graz University of Technology
Klosterwiesgasse 32/2, A-8010 Graz, Austria,
auren@igi.tu-graz.ac.at

[2] Department of Architecture and Architectural Systems, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-8501 Japan
{naoki,kojima,ohsaki}@is-mj.archi.kyoto-u.ac.jp

[3] School of Management, Xi'an Jiaotong University
Xi'an, 710049 P.R.China,
yfxu@xjtu.edu.cn

1 Introduction

Given a convex polygon $\boldsymbol{P}$ in the plane and a positive integer n, we consider the problem of generating a length-uniform triangular mesh for the interior of $\boldsymbol{P}$ using n Steiner points. More specifically, we want to find both a set S_n of n points inside $\boldsymbol{P}$, and a triangulation of $\boldsymbol{P}$ using S_n, with respect to the following minimization criteria: (1) ratio of the maximum edge length to the minimum one, (2) maximum edge length, and (3) maximum triangle perimeter.

These problems can be formalized as follows: Let V be the set of vertices of $\boldsymbol{P}$. For an n-point set S_n interior to $\boldsymbol{P}$ let $\mathcal{T}(S_n)$ denote the set of all possible triangulations of $S_n \cup V$. Further, let $l(e)$ denote the (Euclidean) length of edge e, and let $peri(\Delta)$ be the perimeter of triangle Δ.

Problem 1 $\quad \min_{S_n \subset \mathbf{P}} \min_{T \in \mathcal{T}(S_n)} \max_{e,f \in T} \frac{l(e)}{l(f)}$

Problem 2 $\quad \min_{S_n \subset \mathbf{P}} \min_{T \in \mathcal{T}(S_n)} \max_{e \in T} l(e)$

Problem 3 $\quad \min_{S_n \subset \mathbf{P}} \min_{T \in \mathcal{T}(S_n)} \max_{\Delta \in T} peri(\Delta)$

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 [10], or in view of the intrinsic complexity of Heilbronn's triangle problem, see [14]. For the case of a fixed point set, minimizing the maximum edge length is known to be solvable in $O(n^2)$ time; see Edelsbrunner and Tan [7]. Nooshin et al. [12] developed a potential-based heuristic method for Problem 2, but did not give a theoretical guarantee for the obtained solution.

In this paper, we offer an $O(n^2 \log n)$ heuristic capable of producing constant approximations 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 reasonably large. Our experiments reveal a much better behaviour, concerning the quality as well as the runtime. 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.

We first develop a heuristic we 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 [9] and Feder and Greene [8] developed for clustering problems. We then show how to modify the heuristic, in order to produce a set of n points whose Delaunay triangulation within $\boldsymbol{P}$ constitutes a constant approximation for the problems stated above. Note that the solution we construct is neccessarily a triangulation of constant vertex degree.

Generating triangular meshes is one of the 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 [4,11,6,2,5,15]. 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, e.g., the edge length ratio of its triangles. Chew [6] and Melisseratos and Souvaine [11] construct uniform triangular meshes in the weaker sense that only upper bounds on the triangle size are required. 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 [15] are similar in spirit to our Voronoi insertion method, but do not proceed in a canonical way and aim at different optimality criteria.

A particular application of length-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 [12,13]. 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.

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.

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

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 [9] 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 [15], there called *Delaunay refinement.* Their strategies aim at different quality measures, however, and insertion does not take place in a canonical manner.

The algorithm 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 [9,8] 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. 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.

The observations $d_n = \min\{l(S_n, S_n \cup V), l(V,V)\}$ and $l(V,V) \geq d_n^* \geq d_n$ now imply $d_n = \min\{w_{n-1}(p_n), d_n^*\}$ by (2). But p_n maximizes $w_{n-1}(x)$ for all $x \in \boldsymbol{P}$. So the lemma below (whose proof is omitted) completes the argument.

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

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 not 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 $S_n \cup V$.

3.1 Triangle types. 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 $l(S_j, S_j \cup V) = w_{j-1}(p_j)$.

3.2 Edge length bounds. 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 sufficiently 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 re-running 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 $\mathrm{DT}(S'_n \cup V)$. 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

$$A(\boldsymbol{P}) \leq \pi(n + |V|) - A' \tag{3}$$

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,

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

where A'' denotes the area inside of $\boldsymbol{P}$ which is covered by the circles centered at V. Combining (3) and (4), 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 sufficiently 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. By (3) we have

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

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 $n \geq 2k$.

The following is a main theorem of this paper.

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.

3.3 Computational issues. The time complexity of computing the triangulation T^+ is dominated by Steps 2 and 3 of Algorithm INSERT. 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.

Since we already know the location of the new point p_j in the current Voronoi diagram, the region of p_j can be integrated in time proportional to the degree of p_j in the corresponding Delaunay triangulation, $\deg(p_j)$. We then need to calculate, insert, and delete $O(\deg(p_j))$ weights, and then select the largest one in the next iteration. This gives a runtime of $O(\deg(p_j) \cdot \log n)$ per iteration.

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. Note also that the degree of *all* points in the final triangulation T^+ has to be constant.

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

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 $\boldsymbol{P}$. The proof is omitted in this extended abstract.

Lemma 8. $$d_{long} \geq \tfrac{\sqrt{3}}{2} d_n^*.$$

We strongly conjecture that the statement of Lemma 8 can be strengthened to $d_{long} \geq d_n^*$, which would improve the bounds 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 8 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 8. 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. The easy proof is omitted.

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 $MWT(S^+)$.*

5 Experimental Results

We have performed computational experiments in order to see the effectiveness of the proposed algorithm. For space limitations, we focus on Problem 1 and we only give detailed results for two typical convex polygons. The first polygon is rather fat while the second one is skinny and has a very long edge. The length of the shortest edge in both polygons is roughly equal to d_n for $n = 50$. We have tested the four cases of $n = 50, 100, 200, 300$.

As we described in Section 3, Algorithm INSERT places points on the boundary in the first run, and then is restarted to produce the triangulation. Thereby, the consecutive distance between placed points need to be set to $2d_n \sim 3d_n$ in

points	consec_dist	edge ratio	flips run 1	flips run 2	edge ratio	flips run 1	flips run 2
50	$1 \sim 2$	2.67	81	97	5.08	65	65
	$2 \sim 3$	3.96	81	86	5.43	65	64
100	$1 \sim 2$	2.00	180	214	3.18	147	191
	$2 \sim 3$	2.61	180	174	3.62	147	158
200	$1 \sim 2$	1.96	371	475	2.08	335	427
	$2 \sim 3$	3.23	371	392	2.71	335	343
300	$1 \sim 2$	2.04	567	700	2.1	531	657
	$2 \sim 3$	3.85	567	580	2.5	531	563

order to ensure the desired results theoretically. However, the case of $d_n \sim 2d_n$ is also tested in our experiments. The computational results are summarized below (fat polygon left, skinny polygon right).

We observe that (1) the actual edge length ratio is much better than the worst-case ratio of 6 given by Theorem 3, and (2) the number of flips per insertion of a point is very small (roughly two). Although the example polygons do not satisfy $d_n < l(V,V)/2$ (as required by Theorem 3) we obtained much better ratios. This exhibits the practical effectiveness of the proposed algorithm.

Note also that for a fat polygon (left) the choice of consecutive distance of $d_n \sim 2d_n$ produces a better ratio, while for a skinny polygon (right) the contrary can be observed.

6 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 several criteria has been developed. The basic idea has been to relate the length of triangulation edges to the optimum extreme packing distance. The method 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. We stress that our algorithm works with minor modification for arbitrary polygons with possible holes. Convexity is used solely in the proof of Lemma 8. 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 8 (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 $\boldsymbol{P}$. We pose the question of establishing a version of Lemma 8 for general non-convex polygons, and of improving the respective bound $\frac{\sqrt{3}}{2}$ for the convex case.

Viewed from the point of applications to the design of structures, it is also important to generate a triangular mesh for approximating surfaces such as large-span structures. For this direction, our result concerning Problem 1 can be extended to spherical polygons. More precisely, given a convex polygon whose vertices lie on a hemisphere (or a smaller region of a sphere cut by a plane), the problem is to find a triangular mesh whose points are on the hemisphere that minimizes the objective function of Problem 1. It can be shown that the algorithm obtained by appropriately modifying Algorithm INSERT attains a $3\sqrt{5}$ approximation ratio.

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 fifth author was also supported by NSF of China (No. 19731001) and the Japan Society for the Promotion of Science of Japan. We gratefully acknowledge all these supports.

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. T. Gonzalez, "Clustering to minimize the maximum intercluster distance", *Theoretical Computer Science* 38 (1985), 293-306.
10. D.S. Johnson, "The NP-completeness column: An ongoing guide", *Journal of Algorithms* 3 (1982), 182-195.
11. 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.
12. 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.

13. 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.
14. K.F. Roth, "On a problem of Heilbronn", *Proc. London Mathematical Society* 26 (1951), 198-204.
15. J. Ruppert, "A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation", *Journal of Algorithms* 18 (1995), 548-585.