On Algorithmic Enumeration of Higher-Order Delaunay Triangulations

Yusuke Abe* Yoshio Okamoto†

Abstract

In the pursuit of realistic terrain models, Gudmundsson, Hammar, and van Kreveld introduced higher-order Delaunay triangulations. A usual Delaunay triangulation is a 0-order Delaunay triangulation, thus unique for a non-degenerate point set, while order-k Delaunay triangulations can be non-unique when $k \geq 1$. In this work, we propose an algorithm to list all order-k Delaunay triangulations of a given non-degenerate point set on the plane, when $k \leq 2$, in polynomial time per triangulation. The main technique is the reverse search due to Avis and Fukuda, which exploits the connectedness of a certain graph over all objects to be listed. We also show that the same technique is unlikely to work for $k \geq 3$ by exhibiting an example on which the associated graph is disconnected.

1 Introduction

As a generalization of Delaunay triangulations, Gudmundsson, Hammar, and van Kreveld [6] introduced higher-order Delaunay triangulations. Their work is motivated by realistic terrain modeling, and the subsequent work by de Kok, van Kreveld, and Löffler [4] made a further progress on this problem. Further applications have been found by Benkert, Gudmundsson, Haverkort, and Wolff [2] in constructing minimum-interference networks and by Neamtu [8] in multivariate splines. Recently, constrained higher-order Delaunay triangulations [5, 9] and higher-order Delaunay triangulations of a simple polygon [10] have also been studied.

While a Delaunay triangulation is unique on any non-degenerate set of points, higher-order Delaunay triangulations are not necessarily unique (where a point set is *non-degenerate* if no three points are collinear and no four points are cocircular). Therefore, we may consider several optimization problems over higher-order Delaunay triangulations. However, some of these problems are NP-hard [4, 12], so no polynomial-time algorithm can be expected. There can be lots of approaches to tackle NP-hard problems, and in this work we take an enumerative approach. One of the merits of an enumerative approach is the generality: This can be applied for any kind of objective functions. Also, a partial enumeration can be used to obtain an approximate solution.

Technology for designing enumeration algorithms has been developed for a long time. Among them, the reverse search framework proposed by Avis and Fukuda [1] is quite powerful and suited for enumerating complicated objects. The basic strategy is as follows. First we implicitly define

*Discrete Optimization Laboratory, Department of Information and Computer Sciences, Toyohashi University of Technology, Hibarigaoka 1-1, Tempaku, Toyohashi, Aichi, 441-8580, Japan

†Corresponding author. Graduate School of Information Science and Engineering, Tokyo Institute of Technology, 2-12-1-W8-88, Ookayama, Meguro-ku, Tokyo, 152-8552, Japan. E-mail: okamoto@is.titech.ac.jp. Supported by Global COE Program "Computationism as a Foundation for the Sciences" and Grant-in-Aid for Scientific Research from Ministry of Education, Science and Culture, Japan, and Japan Society for the Promotion of Science.

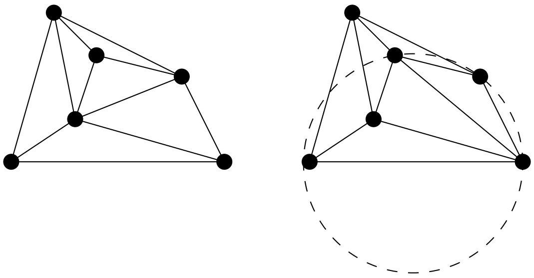

Figure 1: (Left) A Delaunay triangulation. All triangles are Delaunay. (Right) An order-2 Delaunay triangulation. One triangle with its dashed circumscribing disk is of order-2.

a rooted tree on the objects to enumerate. Then, we traverse this tree in a depth-first fashion and output each object when it is visited by the traversal. We try to adapt their strategy for enumerating higher-order Delaunay triangulations of a given non-degenerate set of points.

For $k \leq 2$, we show that the reverse search framework can successfully be adapted, and the order-k Delaunay triangulations can be enumerated in polynomial time per triangulation. In contrast to this positive result, we show that for $k \geq 3$ a natural rooted tree for devising a reverse search algorithm is not well-defined. This indicates that it is difficult to yield an efficient enumeration algorithm for our problem by the reverse search.

2 Preliminaries and statement of our main result

We denote by $\mathbb{R}^2$ the Euclidean plane. A finite set $P \subseteq \mathbb{R}^2$ of points is *non-degenerate* if no three points of P lie on a common line and no four points of P lie on a common circle. A convex hull of a set S of points is denoted by $\overline{S}$. A *triangulation* of P is a decomposition of the convex hull of P into triangles with their vertices in P. We think of a triangulation T of P as a set of triangles forming T. The set of edges of T is denoted by $E(T)$.

Let T be a triangulation of a non-degenerate set P of n points in $\mathbb{R}^2$. A triangle $t \in T$ is called *Delaunay* if the circumscribing disk of t contains no point of P in its interior. If all triangles of T are Delaunay, then T is called *Delaunay.* It is well-known that a Delaunay triangulation uniquely exists for any non-degenerate set of points. Left of Figure 1 shows an example of a Delaunay triangulation.

A triangle $t \in \mathcal{T}$ is of *order-k*, for a natural number k, if the circumscribing disk of t contains at most k points of P in its interior. Note that a Delaunay triangle is of order-0. If all triangles of $\mathcal{T}$ are of order-k, then $\mathcal{T}$ is called an *order-k Delaunay triangulation.* Note that a Delaunay triangulation is an order-0 Delaunay triangulation. Right of Figure 1 shows an example of an order-2 Delaunay triangulation. Note that every triangulation is an order-$(n-3)$ Delaunay triangulation.

The problem we are dealing with in this paper is enumeration. Generally, in an *enumeration problem*, we are given a structure (such as graph, point set) and need to output all objects in the structure that have required properties. The theoretical efficiency of an algorithm for an enumeration problem is measured by running time and space consumption. An enumeration algorithm runs in *polynomial-time delay* if the time spent between two consecutive outputs is bounded by a polynomial of the input size, and runs in *polynomial space* if the working space spent by the algorithm is bounded by a polynomial of the input size.

In this paper, we consider an enumeration of the order-k Delaunay triangulations of a given non-degenerate point set. As our main result, we design an algorithm to enumerate the order-k Delaunay triangulations of a given non-degenerate point set in polynomial-time delay and polynomial space provided that $k \leq 2$.

In the next section, we prove this result with help of the reverse search technique. We show in the final section that the same approach does not work for $k \geq 3$.

3 Proposed algorithm

3.1 Review of the reverse search framework

Basically, we follow the approach of Avis and Fukuda [1] to enumerate the triangulations of a given non-degenerate point set. Therefore, we first explain their algorithm. Let P be a given non-degenerate point set in $\mathbb{R}^2$ and $\mathcal{T}$ be the family of all triangulations of P. In their reverse search framework, we implicitly construct a rooted tree $R(\mathcal{T})$ on $\mathcal{T}$ and traverse it in the depth-first fashion.

An implicit construction of $R(\mathcal{T})$ is given by specifying a unique root triangulation T_r of $R(\mathcal{T})$ and a unique parent triangulation $p(T)$ of a non-root triangulation $T \in \mathcal{T} \setminus \{T_r\}$. We define T_r as a Delaunay triangulation of P, which is unique since P is non-degenerate. To define a parent triangulation, we need to introduce more terminology. Let $T \in \mathcal{T}$ be a triangulation of P and $e = \{u, v\} \in E(T)$ be an edge of T not on the boundary of the convex hull of P. We call e *flippable* if it is the intersection of two triangles $\{u, v, x\}, \{u, v, y\}$ of T and $\mathsf{flip}(T, e) = (T \setminus \{\{u, v, x\}, \{u, v, y\}\}) \cup \{\{u, x, y\}, \{v, x, y\}\}$ is a triangulation of P. We can easily see that e is flippable if and only if $\{u, v, x, y\}$ is a convex quadrilateral (4-gon). The triangulation $\mathsf{flip}(T, e)$ is the outcome of an operation of the *flip* of e. We define a graph $G(\mathcal{T})$ constructed from the family $\mathcal{T}$ of triangulations of P as follows: The vertex set of $G(\mathcal{T})$ is the family $\mathcal{T}$, and two triangulations $T_1, T_2 \in \mathcal{T}$ are joined by an edge in $G(\mathcal{T})$ if and only if $T_1 = \mathsf{flip}(T_2, e)$ for some flippable edge e of T_2. It is well-known that $G(\mathcal{T})$ is connected. In Figure 2, the graph $G(\mathcal{T})$ is shown by solid lines. Our rooted tree $R(\mathcal{T})$ is actually a spanning tree of $G(\mathcal{T})$.

Let $e \in E(T)$ be a flippable edge of a triangulation $T \in \mathcal{T}$ that is the intersection of two triangles $\{u, v, x\}, \{u, v, y\}$ of T. We call e *illegal* if the circumscribing disk of $\{u, v, x\}$ contains y in its interior. Note that this is equivalent to that the circumscribing disk of $\{u, v, y\}$ contains x in its interior. Otherwise, we call e *legal*. When we keep flipping illegal edges, we obtain a sequence of triangulations, and it is well-known that such a sequence does not contain one triangulation multiple times, so eventually we get a triangulation in which all edges are legal. We can see that such a triangulation is indeed a Delaunay triangulation.

Now we define a rooted tree $R(\mathcal{T})$ as follows. First let $<$ be a fixed total order on the family of unordered pairs of points from P. The root of $R(\mathcal{T})$ is a (unique) Delaunay triangulation of P. If $T \in \mathcal{T}$ is not Delaunay, we define the parent of T as $\mathsf{flip}(T, e)$ where e is a (unique) $<$-minimum illegal edge in T. This finishes the definition of $R(\mathcal{T})$, and note that it is well-defined. In Figure 2, the rooted tree $R(\mathcal{T})$ is shown by fat solid lines. From a result by Avis and Fukuda [1], combined with a technique by Uno [11] and Nakano and Uno [7],[1] we can enumerate all the triangulations of a non-degenerate set P of n points in $\mathrm{O}(n)$ time delay and $\mathrm{O}(n)$ space with $\mathrm{O}(n \log n)$ time preprocessing by traversing $R(\mathcal{T})$ from its root.

Note that Bespamyatnikh [3] devised a faster enumeration algorithm for triangulations based on the reverse search technique, but his algorithm uses a different rooted tree. His algorithm introduces a lexicographic order on the family of pairs of points, and the lexicographically smallest triangulation is the root of his search tree. In particular, his algorithm does not use the Delaunay triangulation as a root.

[1] In Uno's paper [11] two techniques are proposed. We always refer to his first technique in this paper, and this is what is used by Nakano and Uno [7].

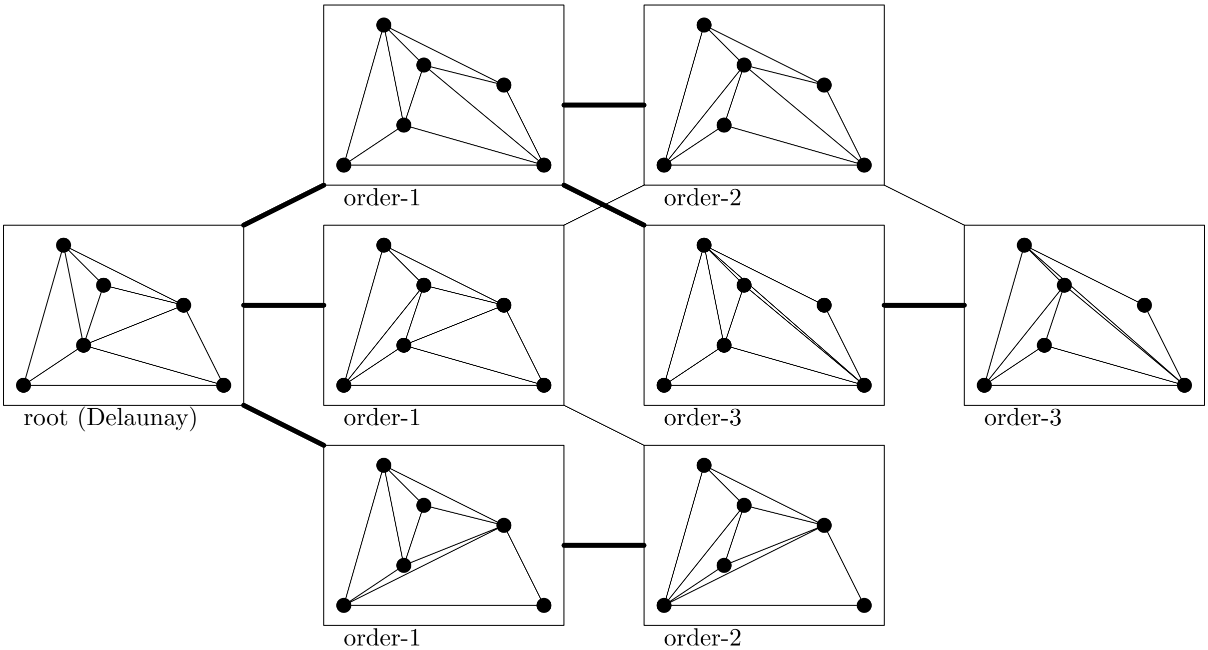

Figure 2: The family of triangulations of a set of six points. There are eight triangulations on this point set, and solid lines between triangulations show the adjacency relationship in $G(\mathcal{T})$. The fat solid lines indicate the adjacency in the rooted tree $R(\mathcal{T})$.

3.2 Enumeration of the order-1 and order-2 Delaunay triangulations

The basic idea of our algorithm is to look at the orders of triangulations in $G(\mathcal{T})$, and if the restriction of $G(\mathcal{T})$ (in the graph-theoretic sense) to the order-k Delaunay triangulations is again connected, then we may perform a reverse search on this restricted graph. Indeed, in Figure 2, this holds for each $k = 0, 1, 2, 3$.

Given a non-degenerate set P of n points in $\mathbb{R}^2$ and a natural number k, define $\mathcal{T}_k$ as the family of order-k Delaunay triangulations of P, and define $G(\mathcal{T}_k)$ as the restriction of $G(\mathcal{T})$ on $\mathcal{T}_k$. The following theorem plays a crucial role in our algorithm.

Theorem 3.1. *For $k \leq 2$, the graph $G(\mathcal{T}_k)$ is always connected.*

To prove Theorem 3.1, we use two lemmas.

Lemma 3.2. *Let T be a triangulation of a non-degenerate set P of points in $\mathbb{R}^2$, and $e = \overline{\{u,v\}} \in E(T)$ be an illegal edge of T, which is the intersection of two triangles $\overline{\{u,v,x\}}$ and $\overline{\{u,v,y\}}$ of T. If the triangles $\overline{\{u,v,x\}}$ are $\overline{\{u,v,y\}}$ of order-k and order-ℓ, respectively, and the triangles $\overline{\{u,x,y\}}$ and $\overline{\{v,x,y\}}$ are of order-k' and order-ℓ', respectively, but not of order-$(k'-1)$ and order-$(\ell'-1)$, respectively, then it holds that $k' + \ell' \leq k + \ell - 2$.*

Proof. By the assumption, the circumscribing disk D_1 of $\overline{\{u,v,x\}}$ contains at most k points from P in its interior, and y is one of those k points. Similarly, the circumscribing disk D_2 of $\overline{\{u,v,y\}}$ contains at most ℓ points from P in its interior, and x is one of those ℓ points. Let D'_1 and D'_2 denote the circumscribing disks of $\overline{\{u,x,y\}}$ and $\overline{\{v,x,y\}}$, respectively. Since the interior of $D'_1 \cup D'_2$ is included in the interior of $D_1 \cup D_2$ and x, y lie on the boundary of $D_1 \cup D_2$, the sum of the number of points in the interior of D'_1 and the number of points in the interior of D'_2 is at most $k+\ell-2$. Therefore, $k'+\ell' \leq k+\ell-2$. See also Figure 3. □

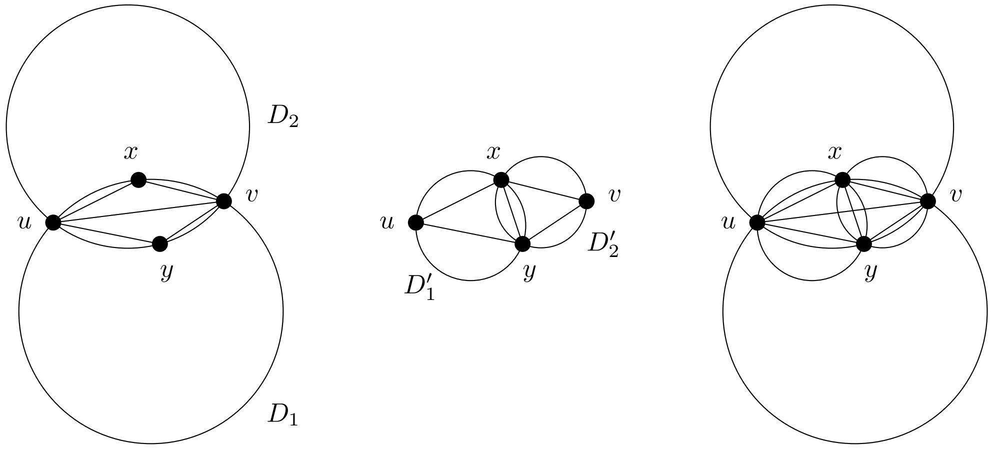

Figure 3: Proof of Lemma 3.2.

Lemma 3.3. *Let $k \leq 2$ and T be an order-k Delaunay triangulation of a non-degenerate set P of points in $\mathbb{R}^2$. If $e \in E(T)$ is illegal, then $\mathsf{flip}(T, e)$ is also an order-k Delaunay triangulation of P.*

Proof. Let an illegal edge $e = \overline{\{u, v\}}$ be the intersection of two triangles $\overline{\{u, v, x\}}, \overline{\{u, v, y\}}$ of T. By the assumption, the triangles $\overline{\{u, v, x\}}, \overline{\{u, v, y\}}$ are of order-k where $k \leq 2$. By Lemma 3.2, the sum of the numbers of points from P in the interior of the circumscribing disk of $\overline{\{u, x, y\}}$ and the number of points from P in the interior of the circumscribing disk of $\overline{\{v, x, y\}}$ is at most $k+k-2 = 2k-2$, which is 0 if k=1 and 2 if k=2. Therefore, both triangles $\overline{\{u, x, y\}}, \overline{\{v, x, y\}}$ are of order-k whenever $k \leq 2$. □

Now we are ready to prove Theorem 3.1.

Proof of Theorem 3.1. Let T be an order-k triangulation of P, where $k \leq 2$. Flipping any illegal edge of T does not increase the order by Lemma 3.3, and as we observed in the previous section successive flips bring T to a (unique) Delaunay triangulation. □

With Theorem 3.1 and modifying the enumeration algorithm for triangulations in the previous section we conclude the following.

Theorem 3.4. *There exists an algorithm to enumerate all order-k Delaunay triangulations of a given non-degenerate set of n points in $\mathbb{R}^2$, if $k \leq 2$. The running time is $\mathrm{O}(n^2)$ per output, with $\mathrm{O}(n \log n)$-time preprocessing, and the memory usage is bounded by $\mathrm{O}(n)$.*

Proof. The algorithm described above does the job. The correctness follows from Theorem 3.1 and the general theory of reverse search [1].

Then, we bound the running time and the memory usage. To begin the reverse search we need to find a Delaunay triangulation, that takes $\mathrm{O}(n \log n)$ time. Then, similarly to a result by Avis and Fukuda [1], we can show that we can find a child triangulation and a unique parent triangulation of any triangulation in $\mathrm{O}(n^2)$ time since we can check that a triangulation obtained by a flip from an order-2 Delaunay triangulation is also order-2 in $\mathrm{O}(n)$ time just by looking at two newly introduced triangles. Therefore, if we combine the argument above with a technique of Uno [11] and Nakano and Uno [7], we obtain the running time $\mathrm{O}(n^2)$ per output, and the memory usage $\mathrm{O}(n)$. □

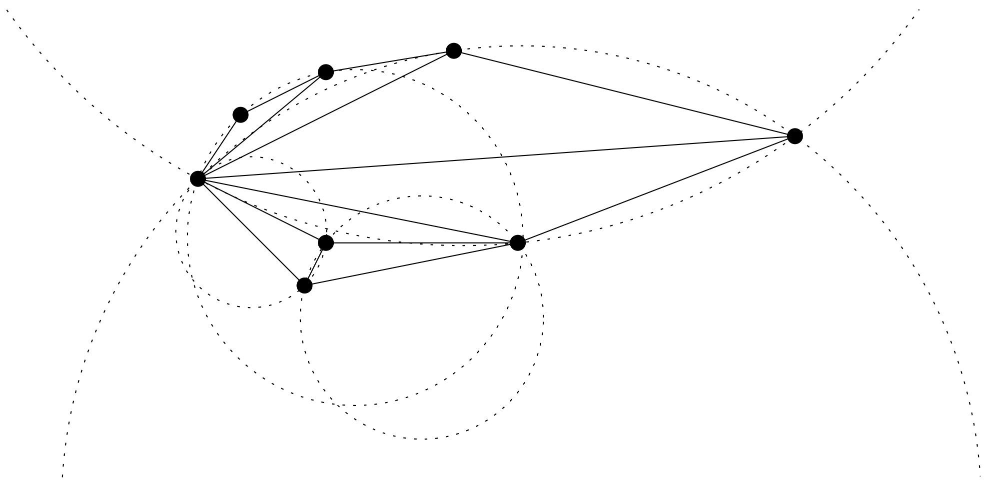

Figure 4: A bad example for $k = 3$.

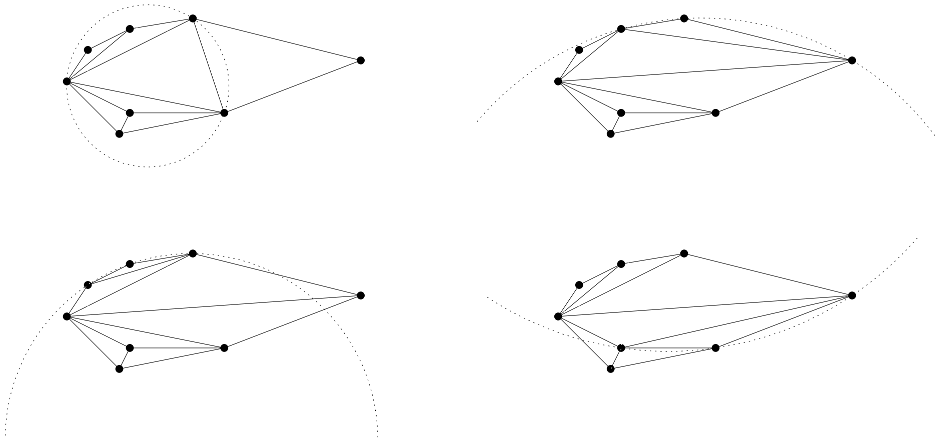

Figure 5: The triangulations resulting from flips in the bad example for $k = 3$.

If we wish to apply the method of Bespamyatnikh [3], then we need to compute the lexicographically minimum (in his sense) order-k Delaunay triangulation. However, it is not clear how to do this in polynomial time.

4 Concluding remarks

The algorithm above does not apply to the case $k \geq 3$. Actually, we exhibit below a non-degenerate set of points for which $G(\mathcal{T}_k)$ is not connected when $k \geq 3$. Thus, it looks difficult to extend the approach in the previous section to that case.

Figure 4 is an example for $k = 3$. The figure shows a triangulation of a set of eight points, and this is of order-3 as illustrated by the set of dotted circles. It has four flippable edges, and Figure 5 shows all resulting triangulations of these flips. We can see that they all contain a triangle whose circumscribing disk contains more than three points in their interiors; Thus these triangulations are not of order-3.

By extending the example in Figure 4, we may show that this sort of bad examples with $k{+}5$ points exist for $k = 4, 5, \ldots, n{-}4$. An example for $k = 4$ appears in Figure 6, but we omit the details here. Therefore, the reverse search approach we adapted in this paper cannot be directly

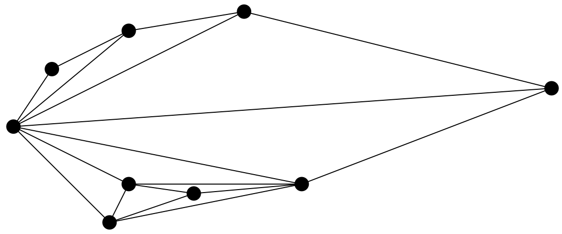

Figure 6: A bad example for $k = 4$.

applied to higher-order Delaunay triangulations. This however does not exclude the possibility of efficient enumeration of these triangulations either. This remains an unsolved problem.

Acknowledgments The authors are grateful to Toshihiro Fujito and the members of Discrete Optimization Laboratory, Department of Information and Computer Sciences, Toyohashi University of Technology for constructive suggestions.

References

[1] D. Avis, K. Fukuda. Reverse search for enumeration. Discr. Appl. Math. **65** (1996) 21–46.

[2] M. Benkert, J. Gudmundsson, H.J. Haverkort, A. Wolff. Constructing interference-minimal networks. Proc. SOFSEM 2006, pp. 166–176.

[3] S. Bespamyatnikh. An efficient algorithm for enumerating triangulations. Comput. Geom. **23** (2002) 271–279.

[4] T. de Kok, M. van Kreveld, M. Löffler. Generating realistic terrains with higher-order Delaunay triangulations. Comput. Geom. **36** (2007) 52–65.

[5] J. Gudmundsson, H.J. Haverkort, M. van Kreveld. Constrained higher order Delaunay triangulations. Comput. Geom. **30** (2005) 271–277.

[6] J. Gudmundsson, M. Hammar, M. van Kreveld. Higher order Delaunay triangulations. Comput. Geom. **23** (2002) 85–98.

[7] S.-I. Nakano, T. Uno. Generating colored trees. Proc. 31st WG (2005) pp. 249–260.

[8] M. Neamtu. Delaunay configurations and multivariate splines: A generalization of a result of B.N. Delaunay. Trans. Amer. Math. Soc. **359** (2007) 2993–3004.

[9] R.I. Silveira, M.J. van Kreveld. Towards a definition of higher order constrained Delaunay triangulations. Proc. CCCG 2007, pp. 161–164.

[10] R.I. Silveira, M.J. van Kreveld. Optimal higher order Delaunay triangulations of polygons. Proc. LATIN 2008, pp. 133-145.

[11] T. Uno. Two general methods to reduce delay and change of enumeration algorithms. NII Technical Report **NII-2003-004E** (2003), 10 pages.

[12] M.J. van Kreveld, M. Löffler, R.I. Silveira. Optimization for first order Delaunay triangulations. Proc. WADS 2007, pp. 175–187.