New Greedy Triangulation Algorithms (Extended Abstract)

Oswin Aichholzer
Franz Aurenhammer
Institute for Theoretical Computer Science
Graz University of Technology
Klosterwiesgasse 32/2, A-8010 Graz, Austria
e-mail: {oaich,auren}@igi.tu-graz.ac.at

Günter Rote
Institut für Mathematik
Technische Universität Graz
Steyrergasse 30
A-8010 Graz, Austria
e-mail: rote@ftug.dnet.tu-graz.ac.at

Yin-Feng Xu
School of Management
Xi'an Jiaotong University
Xi'an Shaanxi, 710049, PRC

1 Introduction

The classical greedy triangulation (GT) of a set S of n points in the plane is the triangulation obtained by starting with the empty set (of edges) and at each step adding the shortest compatible edge between two of the points of S, where a compatible edge is defined to be an edge that crosses none of the previously added edges.

One use of the greedy triangulation is as an approximation to the minimum weight triangulation (MWT). The MWT of a point set minimizes the total length of all edges, where the length of an edge is the Euclidean distance between its two endpoints. It arises in numerical analysis [Li, Ll, PS], numerical approximation of bivariate data [Yo], or in the reconstruction of surfaces from contours [WA].

Computing a minimum-weight triangulation is an important and interesting problem, whose complexity status is unknown. It is one of the still open problems listed at the end of Garey and Johnson's book about NP-completeness [GJ]. We therefore seek efficiently computable *approximations* to the MWT.

Although neither the GT nor the Delaunay triangulation (DT) yields the MWT [MZ, Ll], the GT appears to be the better of the two in approximating the MWT. In fact, for convex polygons the GT approximates the MWT to with a constant factor while the DT can be a factor of $\Omega(n)$ larger [LL]. For general point sets a factor of $\Omega(\sqrt{n})$ is known [K, Lev] as a lower bound for the GT.

In this paper we use the greedy method as a general concept to compute a triangulation of a planar point set. As mentioned in [AART] we use either edges or triangles as basic objects. Furthermore we give different variants to compute the weight of the objects, either in a static or dynamic way, leading to a total of 156 different greedy triangulation algorithms. We investigate these algorithms in their quality of approximating the MWT for sample point sets. We check whether there are methods superior to others for all (or at least most) of these sets, which are the 'best' for uniformly distributed point sets, or which do avoid worst cases. We used a generic implementation and thus did not optimize the different algorithms.

2 The new methods

The general greedy triangulation procedure may be described as follows: First sort the 'objects' by their weight in non-decreasing order. Then start to insert the objects in this order into the (initially empty) triangulation, where only compatible objects are inserted. Repeat the last step until all compatible objects are inserted, i.e. until a triangulation is obtained. To avoid ambiguity and for simplicity, we assume all weights to be pairwise different.

To clarify this description just consider the 'usual' greedy triangulation. The objects are edges of the point set, the weight is their Euclidean length and compatible means that the edge to be inserted does not intersect any previously inserted edge.

We now consider different possibilities for alternative greedy algorithms.

First, objects may either be edges or empty triangles (triangles with no inner points) of S. For both types the same interpretation of 'compatible' is used, meaning intersection-free in their interior. Note that two triangles having one edge in common are compatible in this sense.

Next we distinguish between the size $s(a)$ and the weight $w(a)$ of an object a. The size of an object is a measure of the object itself and does not depend on other objects generated by the point set. For edges we just take their Euclidean length as their size:

1. $s(a) =$ length of edge a

For triangles we have different possibilities:

1. $s(a) =$ perimeter of a
2. $s(a) = \frac{\text{max edge length of } a}{\text{min edge length of } a}$
3. $s(a) =$ area of a
4. $s(a) = \frac{\text{max angle of } a}{\text{min angle of } a}$
5. $s(a) =$ radius of mindisc (smallest enclosing disc) of a
6. $s(a) =$ radius of incircle of a
7. $s(a) =$ radius of the circumcircle of a

The weight of an object is derived from its size and possibly also from the sizes of the objects intersecting it. Let a be an object and I_a be the set of objects intersecting it. E.g. a is an edge and I_a is the set of crossing edges, or a is a triangle and I_a is the set of overlapping triangles. Note that I_a does not include a itself. We use the following different methods to compute the weigth $w(a)$:

1. $w(a) = s(a)$
2. $w(a) = \frac{1}{\max\limits_{a' \in I_a} s(a')}$
3. $w(a) = \frac{1}{\min\limits_{a' \in I_a} s(a')}$
4. $w(a) = \frac{1}{\sum\limits_{a' \in I_a} s(a')}$
5. $w(a) = \frac{1}{\operatorname{mean}\limits_{a' \in I_a} s(a')}$
6. $w(a) = \frac{s(a)}{\max\limits_{a' \in I_a} s(a')}$
7. $w(a) = \frac{s(a)}{\min\limits_{a' \in I_a} s(a')}$
8. $w(a) = \frac{s(a)}{\sum\limits_{a' \in I_a} s(a')}$
9. $w(a) = \frac{s(a)}{\operatorname{mean}\limits_{a' \in I_a} s(a')}$

A. $w(a) = \frac{1}{|I_a|}$

B. $w(a) = \frac{s(a)}{|I_a|}$

The first weight is just the size of the object as used for the usual greedy method. Note that for weights $2) - B)$ we set $w(a) = 0$ if I_a is empty, witnessing that we have an unavoidable object.

There are two different ways to compute the set I_a: the static and the dynamic method. In the static method we precompute all sets I_a and the weights of all objects which can be derived from the point set. Thus the weights do not change during the insertion phase.

In the dynamic method we recompute the sets I_a and thus the weights after each insertion step if necessary: The sets I_a only contain compatible objects, i.e. objects which do not intersect any previously inserted objects and thus are still 'relevant'. This means that after inserting an object we have to remove all its intersecting objects and to update all involved sets I_a and associated weights. The dynamic method seems to be more sophisticated and we expect it to produce better results. For instance, a long crossing edge does only affect the weight of shorter edges as long as it is possible to insert the long edge into the triangulation. In other words: why should an object affect other objects when we already know that it cannot be part of the triangulation?

Counting the number of possible greedy procedures we have 11 static and 10 dynamic methods to compute the weights, one method to compute the size of an edge and 7 variants to compute the size of a triangle. Except method A) we can combine all weights with all sizes, leading to a total of 156 different greedy triangulation algorithms.

To simplify notation we give a four character identification for the different methods. The first character shows if we use edges (E) or triangles (T) as objects, the next character stands for the static (S) or dynamic (D) method. The following two characters give the number of the formula to compute the size of the objects and their weight, respectively. Thus for the usual greedy triangulation algorithm we have 'ES11'. Note that not all possible combinations are different, as for example 'TD11' which is just the same as 'TS11'.

3 Preliminary Results and Discussion

Because of the huge number of different methods our investigations and the experiments are still going on. Thus for the extended abstract we just mention some of the preliminary results we got so far.

First let us compare a static edge method (ES11, 'usual' greedy) and a triangle method (TS11), both using the edge length of objects as size and weight.

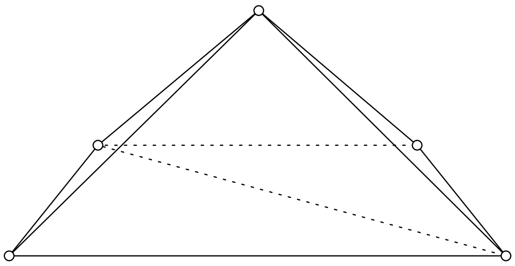

Figure 1: Optimal triangulation of a pentagon (TS11, solid) and the GT (ES11, dotted).

In Figure 1 a well known five point example shows that the GT (ES11) does not produce the MWT in general. Here method TS11 inserts first the small triangles on the left and right side, thus attaining the optimal triangulation. On the other hand ES11 is known to be local optimal while TS11 fails to be, even in the four point example of Figure 2. Thus non of these two methods is superior to the other on all point sets.

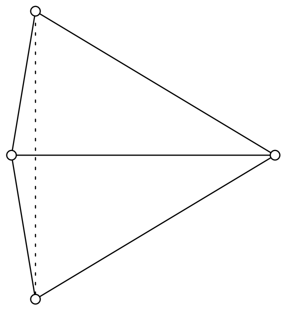

Figure 2: GT of a quadrilateral (ES11, solid) and the non local optimal TS11 (dotted).

To get a ranking we first need an appropriate cost function: For a given point set S let T be the set of triangulations produced by the mentioned greedy triangulation methods. Let $w_{max}(S)$ be the length of the longest and $w_{min}(S)$ the length of the shortest triangulation in T. To each method M we assign the cost $c_M(S) = \frac{w_M(S)-w_{min}(S)}{w_{max}(S)-w_{min}(S)} \in [0,1]$ where $w_M(S)$ is the length of the triangulation produced by method M. Thus $c_M(S) = 0$ means that method M produces (one of) the shortest triangulation in T, while $c_M(S) = 1$ shows that it achieves the upper bound. With c_M we denote the average of $c_M(S)$ over all investigated point sets S for a

Method M	Average Cost c_M
ED17	0.00041
ES11	0.00041
TS51	0.00106
TD57	0.00106
TS59	0.00135
TS57	0.00140
ES17	0.00286
TS56	0.00298

Table 1: The eight best methods for 25 uniformly ditributed point sets with costs.

Method M	Average Cost c_M
ED16	0.01104
ED17	0.02736
ES11	0.02736
ES16	0.02788
TS56	0.02992
TD56	0.03327
ES17	0.03745
TD19	0.03832

Table 2: The eight best methods for 40 'special' point sets with costs.

method M. We use this average cost for ranking the methods.

In a first phase we computed the cost of every method for 25 uniformly distributed point sets with up to 56 points. (This limitation steems from the huge number of methods and the generic and thus slow implementation; we plan to do experiments on larger point sets in the future). Table 1 shows the eight best methods with their costs. On the average their length is less than 0.3% of the longest triangulation of T and 61 of the methods have an average length of less than 1% of w_{max}. This shows that for uniformly distributed point sets most of the greedy methods behave quiet well, as was already known for the GT.

To get a better classification in a second phase we added 15 special point sets, e.g. the $\sqrt{n}-$lower bound example of Levcopoulos [Lev] or the examples of Figures 1 and 2. This gave a much clearer distinction of the methods and led to the ranking shown in Table 2. Several observations may be made from this table. First of all, no mentioned greedy method is optimal for all point sets, i.e. produces always the MWT. The GT (ES11)

Group	Average Cost	Best/Groub	Cost of Best
EXXX	0.18486	ED16	0.01104
TXXX	0.28242	TS56	0.02992
XSXX	0.29056	ES11	0.02736
XDXX	0.24736	ED16	0.01104
ESXX	0.23825	ES11	0.02736
EDXX	0.12479	ED16	0.01104
TSXX	0.29803	TS56	0.02992
TDXX	0.26486	TD56	0.03327
XXXX	0.27023	ED16	0.01104

Table 3: Group of methods with 'Leader' and cost.

is among the best, witnessing its well known average behaviour. The method ED16 seems to be very stable against 'bad' constructed examples. While e.g. the GT has now approx. 65 times of its previous costs, so still beeing third, ED16 has increased its cost by less than 50% (beeing only no. 13 for random point sets).

In Table 3 we compared groups of methods to get an idea which sizes, weights etc. are good. In the identification used, an X means any valid character, i.e. EXXX are all edge-based methods. We make the following observations: Edge-based methods are significantly better than triangle-based methods. Dynamic methods prevail static ones, but only about 15%, even less for triangle-based methods. Only for edge-based triangulations they give a reasonable improvement. Most of the methods produce short triangulations and only a few behave bad, as can be seen from the small average cost of 0.27023 for all triangulations of T. It is also interesting to note that, for triangles size choice 5 gives the best results in average, while for the weight choice 6 is best not only for triangle- but also edge-based methods. Thus up to now dynamic edge-based methods seem to be preferable, especially ED16.

To refine our investigations it is necessary to consider more and larger point sets. To this end we will concentrate only on the promising subset of methods, e.g. all methods with cost less than 0.1 for the second phase (about 45). This will allow us to get a better distinction of the methods and to get some ideas how a 'optimal' algorithm should look like.

Acknowledgements: We would like to thank J. Lackinger for implementing the generic algorithm.

References

[AART] O. Aichholzer, F. Aurenhammer, G. Rote, M. Taschwer, *Triangulations intersect nicely*, Proc. 11th Ann. Symp. on Computational Geometry, Vancouver, British Columbia, June 1995.

[GJ] M. Garey and D. Johnson, *Computers and Intractability. A Guide to the Theory of NP-completeness*, Freeman, 1979.

[K] D. Kirkpatrick, "A note on Delaunay and optimal triangulations." *Information Processing Letters* **10** (1980) 127–128.

[Lev] C. Levcopoulos, "An $\Omega(\sqrt{n})$ Lower Bound for the Nonoptimality of the Greedy Triangulation." *Information Processing Letters* **25** (1987) 247–251.

[Li] A. Lingas, "On Approximating Behavior and Implementation of the Greedy Triangulation for Convex Planar Point Sets." *Proceedings of the Second Annual Symposium on Computational Geometry* (1986) 72–79.

[LL] C. Levcopoulos and A. Lingas, "On Approximating Behavior of the Greedy Triangulation for Convex Polygons." *Algorithmica* **2** (1987) 175–193.

[Ll] E. L. Lloyd, "On triangulations of a set of points in the plane." *Proceedings of the 18th IEEE Symp. Foundations of Computer Science* (1977) 228–240.

[MZ] G. Manacher and A. Zobrist, "Neither the greedy nor the Delaunay triangulation of the planar set approximates the optimal triangulation." *Information Processing Letters* **9** (1979) 31–34.

[PS] F. Preparata and M. Shamos, "Computational Geometry: an Introduction." Springer-Verlag, 1985.

[WA] Y. F. Wang and J. K. Aggarwal, "Surface reconstruction and representation of 3-D scenes." *Pattern Recognition* **19** (1986) 197–207.

[Yo] P. Yoeli, "Compilation of data for computer-assisted relief cartography." In *Display and Analysis of Spatial Data*, J. C. Davis and M. J. McCullagh, editors, John Wiley & Sons, NY (1975).