A Branch-and-Cut Approach for Minimum Weight Triangulation

Yoshiaki Kyoda[1], Keiko Imai[2], Fumihiko Takeuchi[1] and Akira Tajima[3]

[1] Department of Information Science, University of Tokyo
[2] Department of Information and System Engineering, Chuo University
[3] IBM Tokyo Research Laboratory

Abstract. This paper considers the problem of computing a minimum weight triangulation of n points in the plane, which has been intensively studied in recent years in computational geometry. This paper investigates a branch-and-cut approach for minimum weight triangulations. The problem can be formulated as finding a minimum-weight maximal independent set of intersection graphs of edges. In combinatorial optimization, there are known many cuts for the independent set problem, and we further use a cut induced by geometric properties of triangulations. Combining this branch-and-cut approach with the β-skeleton method, the moderate-size problem could be solved efficiently in our computational experiments. Polyhedral characterizations of the proposed cut and applications of another old skeletal approach in mathematical programming as the independent set problem are also touched upon.

1 Introduction

A triangulation of a planar point set S is a subdivision of the convex hull, denoted by CH(S), of S into triangles, and is a maximal straight line plane graph whose vertices are the points of S. Triangulations have many applications in computational geometry and related fields. What kind of triangulation is optimal depends on applications. For instance, we do not want flat obtuse triangles for finite element triangulation mesh. Many kinds of optimal triangulation have thus been investigated [3, 8]. Optimization criteria for which efficient algorithms are known include maximizing the minimum angle (Delaunay triangulation), minimizing the maximum angle, minimizing the minimum angle maximizing the minimum height, and minimizing the maximum edge length, etc.

The most longstanding open problem in computational geometry is the minimum weight triangulation (MWT in short), in which the criterion is the sum of edge length. MWT is included in Garey and Johnson's list of problem as neither known to be NP-complete, nor known to be solvable in polynomial time, though for a point set which forms a convex polygon, dynamic programming can compute the MWT in $O(n^3)$ time where n is the size of the point set.

The apparent difficulty of the problem suggests that approximation algorithms should be considered. It was thought that the Delaunay triangulation and the Greedy triangulation (GT in short) approximate the MWT well. Construction and properties of these have been most studied of all (see references in [3, 6, 7] and also [5, 15]). These approximate algorithms produce good triangulations on the average, and also several theoretical analyses for these approximate algorithms are performed (see [11]).

On the other hand, there have been demonstrated that a large subgraph, called skeleton, contained in any minimum weight triangulation can be computed in a polynomial time by making use of plane-geometric properties, and this works quite well for points uniformly distributed in the square, etc., by combining it with dynamic programming. See [6, 5, 7, 15]. In this approach, as far as the number of connected components of the skeleton is bounded, the problem can be solved in a polynomial time, but there exist cases such that this number is not bounded by a constant. For such a case, a new approach would be necessary.

In this paper, we have adopted another approach based on the paradigm of branch-and-cut in combinatorial optimization. This paradigm has been demonstrated to be powerful enough to solve large-scale optimization problems such as the traveling salesman problem (TSP) (e.g., see [1]). Study of solving TSP faster is to find good cutting planes efficiently. In our implementation, we made use of a skeleton [9, 4] which is always contained in MWT, and also many kinds of cutting planes, especially convex polygon cuts making use of geometry. Though with only cutting planes of ours, an MWT cannot be necessarily obtained, we can find an MWT of point sets of over one hundred points without branching and, even when cutting plane method can not construct MWT, a much less number of backtrackings are needed than the branch-and-bound method.

Polyhedral characterizations of the proposed convex polygon cut and applications of another old skeletal approach via network flow in mathematical programming as the independent set problem are also touched upon.

2 Problem formulation

The minimum weight triangulation problem can be viewed in various ways. This paper regards finding a triangulation of a point set as finding a maximum independent set of the intersection graph of the straight line complete graph of the given point set. An edge in the original graph is a vertex in the corresponding intersection graph and those vertices are connected if they cross each other in the original graph.

For each edge e_i, we assign a variable x_i. x_i is set to 1 when e_i is used in the triangulation, and 0 otherwise. Denote by c_i the length of edge e_i, and by M the number of edges used in triangulation. An edge subset of the complete graph of the n points such that no two edges intersect with each other and the number of edges is M, a constant for any triangulation. Hence, the MWT problem is formulated as follows.

$$\min \left\{ \sum_i c_i x_i \;\middle|\; x_i \in \{0,1\}, \quad x_i + x_j \leq 1 \text{ (} i,j \text{ s.t. } e_i \text{ and } e_j \text{ cross)}, \quad \sum_i x_i = M \right\}$$

In general, such Integer Program (IP) is not easy to solve and some devices are needed to reduce computational time. Here we consider solving this problem by branch-and-bound and branch-and-cut paradigms. In both methods, we utilize Linear Program (LP) obtained by relaxing the condition $x_i \in \{0,1\}$ into $0 \leq$

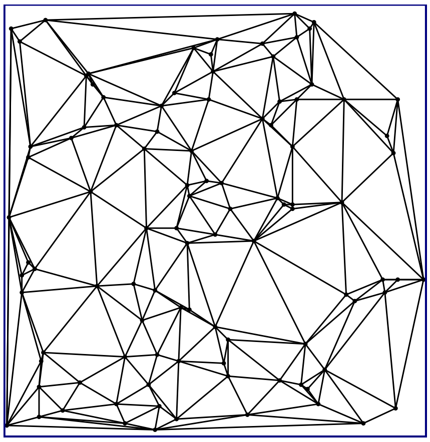


Fig. 1. Minimum weight triangulation of a set of 100 points

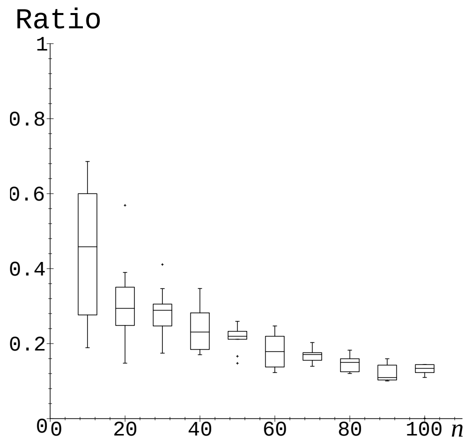


Fig. 2. Effects of β-skeletons: Ratio of #essential var. to #nontrivial var.

$x_i \leq 1$, called "relaxed problem" of IP or "relaxed MWT". As for our branch-and-bound results, see [10]. In this paper we concentrate on the branch-and-cut paradigm.

2.1 Exploiting β-skeleton

There have been investigated intensively a subgraph which is always included in the MWT. In this paper, we used the β-skeleton [4, 6] among them. The LMT-skeleton [7, 5] has been demonstrated more powerful in most cases, and, by using it, further improvement would be achieved. By using the β-skeleton, we can fix variable x_i of edge e_i in the skeleton to 1, and further variable x_j of edge e_j intersecting some edge in the subgraph to 0.

Figure 2 shows the ratio of the number of essential variables to the total number of edges minus the number of boundary edges of the convex hull, for n points uniformly distributed in the square. As is seen from this figure, the size of the LP is reduced greatly by making use of this subgraph.

3 Branch-and-cut for MWT

The branch-and-cut algorithm solves Integer Programming (0-1 programming) by means of Linear Programming as follows. First it solves the related LP of the original IP. If the solution is integral, it is the solution of IP. If not, it appends to the LP some cutting planes which is not satisfied by the current solution and is guaranteed to be satisfied by the optimal integral solution. Then the algorithm solves the LP again, and iterates this process until integral solution is obtained or no cutting planes among prescribed types of candidate cuts violate the solution. In the latter case, the algorithm branches into two cases. One of variables, which are neither 0 nor 1 in the current solution, is set to 0 in one case, and to 1 in the other case. For each case, the algorithm proceeds to solve an LP obtained by fixing the value of the branching variable, and to append cutting planes as before. Then the minimum value of the objective function with integral solution is finally obtained.

In the problems such as TSP or MWT, each edge is assigned to one variable and branching corresponds to adopting the edge and discarding the edge.

3.1 Cutting planes

In order to reduce the total computation time, we have to append cutting planes with good properties and the number of them should be as small as possible. Also we have to reduce the total number of cutting planes by omitting useless cutting planes.

In this section, we first describe typical cutting planes for the independent set problem, and then, by making use of the special structure of triangulations, propose a new cut, called convex polygon cut.

Dominance condition In the intersection graph, to obtain a maximum independent set of size M, for each vertex v_i and a set δv_i of its adjacent vertices v_j, one of v_i and δv_i should be used, i.e.,

$$\text{Dominance cut}: \quad x_i + \sum_{v_j \in \delta v_i} x_j \geq 1.$$

There are $O(n^2)$ inequalities to be satisfied for a set of n points, and this number is relatively smaller with those of other conditions. In some instance, addition of dominance conditions made a great progress and this condition is indispensable in other instances.

Clique condition In the intersection graph, only at most one of the nodes in a clique can be chosen for an independent set. Hence, for a clique (complete subgraph) S of vertices in the intersection graph, the following holds.

$$\text{Clique cut}: \quad \sum_{x_i \in C} x_i \leq 1$$

Ideally, we may impose this clique condition for each maximal clique of the intersection graph, or for every k-clique for some k. However, it takes too much time to generate all maximal cliques. In our implementation, cliques of size 3 and 4 are used. Although there are many cliques in the intersection graph, we have an example that clique cuts are not so effective by themselves in a sense that even with these cuts the relaxed LP solution remains unchanged.

Odd-cycle condition For an odd cycle C' of $2k+1$ vertices ($k \geq 1$) in the intersection graph, only at most k vertices out of C' can be used in independent sets, i.e.,

$$\text{Odd-cycle cut}: \quad \sum_{v_i \in C'} x_i \leq k.$$

The 3-cycle condition is equivalent to 3-clique condition. It is experimentally experienced that omission of conditions for odd cycle of size more than 5 are rarely violated by the solution of LP for MWT. In other words, we often encounter such a point set that 5-cycle condition are very efficient.

Convex polygon condition The above cuts are well-known for the independent set problem. Now we propose a new cut using the structure of planar triangulations.

Theorem 1. *We set P a point set that forms a convex polygon and set I the points inside P. Also we set $V = P \cup I$. For variables that correspond to edges which is not on* CH(V) *and whose both endpoints are in V (we represent by $x_i \in V$ that edge e_i is in the region* CH(V)*), m is the maximal number of edges in triangulation of V minus* $|\mathrm{CH}(V)|$*), the following inequality*

$$\text{Convex polygon cut}: \quad \sum_{x_i \in V} x_i \le m$$

must be satisfied by the variables corresponding to minimum weight triangulation.

All the proofs are omitted in this version due to the space limitation.

Unsolvable cases Even with these cuts, there were cases whose MWT cannot be found. Fig.3 depict that the basic cuts except convex polygon cuts are not sufficient to produce an integral solution to the relaxed LP.

Further, even with convex polygon cuts, the same holds (see Fig.4). The point set in Fig.4 is on an ellipse whose aspect ratio is $\frac{3}{4}$. Edges whose corresponding variables are not 0 in an optimal solution of relaxed LP are drawn in Fig.4. Denoting by $x_{i,j}$ the edge connecting point i and point j, variables for edges on the convex hull of the point set are set 1, and yet $x_{1,4} = x_{5,8} = \frac{2}{3}$ and variable for other diagonals are $\frac{1}{3}$. We have not devised practical cutting planes to avoid such situations.

From the viewpoint of polyhedral combinatorics, merits of convex polygon cuts over the basic cuts, which will be theoretically shown in section 5. Here, it should be noted that, even for 7-gon, the independent set polytope (see that section) is nonintegral, while for 6-gon, it is integral, which cannot be achieved by only the basic cuts.

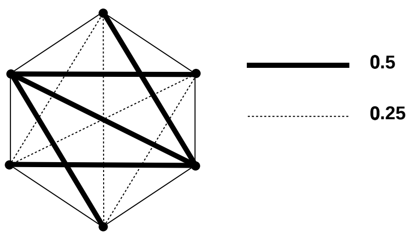


Fig. 3. Edges avoiding clique and cycle conditions

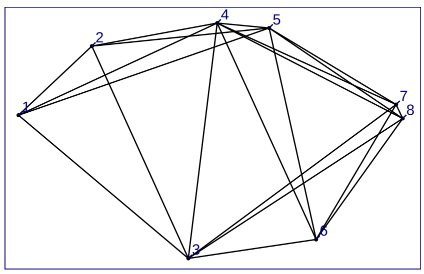


Fig. 4. Unsolved situation

How to find useful cuts To find the condition violated by the solution, we first construct intersection graph of the graph and divide it into connected components. Node adjacent to no nodes is discarded. We have only to consider this condition for the cycle in each component to reduce total calculation time. Because if a variable in some 5-cycle is 1, variables corresponding to the adjacent two node are set 0. Sum of the remaining two variables are at most 1 and then the sum of variables in the cycle is at most 2. Assumption of one of variables corresponding to the node in cycle leads to the same conclusion.

However, enumeration of all the inequalities violated by the solution is not necessarily a good solution to avoid needless inequalities. We shall show how to do this in the later section.

3.2 Strategies in our branch-and-cut

Reducing the number of cutting planes is a key to reduce total time of calculation. So far, we have devised the following two strategies.

(1) Now, we do not apply dominance condition or clique condition at first and treat them as special cases of cycle condition and convex polygon condition. In our first implementation of branch and cut, we first used all the dominance conditions because the total number is not so large. Actually there are only $O(n^2)$ cutting planes in comparison with $O(n^4)$ inequalities of crossing condition. However, each size of dominance condition is so long that total amount of dominance condition is not negligible.

(2) We do not try to find all the odd cycle conditions, even 5-cycle conditions. If the size of any connected component of rounded-up intersection graph (intersection graph of edges whose variables are more than 0) is large, say 30, there might be thousands of cutting planes unsatisfied by the solution of the step. Even worse, some of the violated inequalities are weakly violated, like $x_i + x_j + x_k + x_l + x_m = 2.01$, and improvement for them does not give a big improvement of the solution. In such cases, we had better try convex polygon condition first. When the rounded-up intersection graph become sparse in the connected component after applying the convex polygon ones, we can append much smaller number of cycle conditions.

4 Experimental results

4.1 Implementation

Our experiments were done on SparcStation 20 with 128Mb RAM. Times are all measured on this machine. As for a LP code, we have used an ftp-able code, written in C, called "`lp_solve`" [2], which uses a simplex method. We have also tested some interior-point codes for linear programming to our problem, and it is observed that the interior-point codes solve the problem faster for large cases in our preliminary experiments.

There are the following limitations of our implementation. For convenience we have assumed that no three points are collinear. Furthermore, in the implementation of this sort of search, arithmetic operation of high accuracy is indispensable. However, most arithmetic operations are done by floating-point computation and arithmetic errors such as rounding error or cancellations of significant digits might be caused, though no fatal errors have occurred so far.

4.2 Computation time

In Fig.5, computational results for uniformly distributed points are given. For 100 points uniformly distributed in the square, our program could find an optimal solution within from 30 minutes to 3 days depending on the number of branchings for each instance. By incorporating that the subproblems are divided into much smaller problems in practice and other possible improvements mentioned in this paper, the speed of the current code will be greatly improved.

5 Geometric aspects of the cutting planes

We used several cutting planes to solve the MWT problem. Among those were clique cuts, odd-cycle cuts and convex polygon cuts. In our approach, we characterized triangulations as maximum independent sets of the intersection graph of the edges. In this section, we consider the relation between these three types of cuts and the polytope of the independent sets.

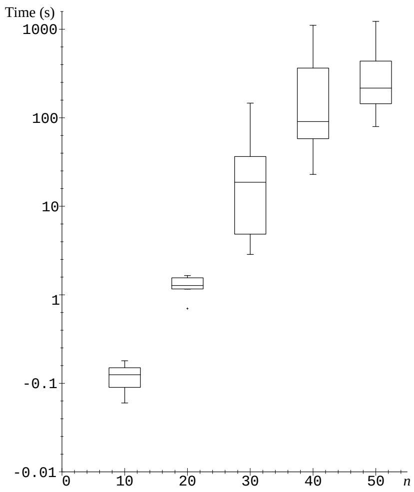
We were given n points $S = \{p_1, \dots, p_n\}$ in the plane. Here, we denote an edge connecting points p_i and p_j by e_{ij}. Let $E = \{e_{ij} : 1 \leq i < j \leq n\}$ be the set of edges. The intersection graph of the edges was a graph having edges E as vertices, and pairs of vertices corresponding to pairs of intersecting edges in E were connected by edges in this intersection graph.

Given an intersection graph, its independent set polytope P_{ind} is the convex hull of the incidence vectors of the independent sets of the intersection graph. The maximal independent set polytope $P_{\max}$ is the convex hull of the incidence vectors of the maximal independent sets. Since these two are 0-1 polytopes, incidence vectors of independent sets and maximal independent sets form exactly the vertices of P_{ind} and $P_{\max}$. These two polytopes can be defined for any graph.

Our intersection graph was the intersection graph of edges of a point configuration S in plane. Euler's formula implies that any spanning triangulation (i.e. a triangulation in which the points in S become the set of 0-dimensional simplices) has a constant number of triangles and edges. So, all maximal independent sets have the same cardinality, and they are maximum. Each maximum independent set corresponds to a triangulation. We defined M to be the number of edges in a triangulation. The inequality $\sum_{e_{ij} \in E} x_{ij} \leq M$ is valid on P_{ind}, and $P_{\text{ind}} \cap \{\boldsymbol{x} : \sum_{e_{ij} \in E} x_{ij} = M\} = P_{\max}$. So $P_{\max}$ forms a face of the polytope P_{ind}.

Let $P_{\text{rel}} = \{\boldsymbol{x} : \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{1},\ x_{ij} + x_{kl} \leq 1 \text{ (for } e_{ij} \text{ and } e_{kl} \text{ intersecting)}\}$. This polytope can be defined for any graph. Each inequality is corresponding to a 2-clique in the intersection graph. The lattice points in this polytope are the incidences vectors of independents sets, or the vertices of P_{ind}. So, $P_{\text{ind}} \subset P_{\text{rel}}$.

The relaxed MWT in subsection 2 is the point set $P_{\text{rel}} \cap \{\boldsymbol{x} : \sum_{e_{ij} \in E} x_{ij} = M\}$. The lattice points here are the incidence vectors of the maximum independent sets, or the vertices of $P_{\max}$. So, $P_{\max} \subset P_{\text{rel}} \cap \{\boldsymbol{x} : \sum_{e_{ij} \in E} x_{ij} = M\}$.



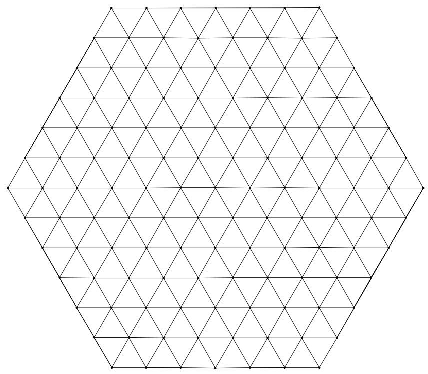


Fig. 5. Computational times for 10 sets of n points uniformly distributed in the square for $n = 10, 20, 30, 40, 50$

Fig. 6. Case of triangular grid: Application of Nemhauser and Trotter's 'skeletal' approach

Convex polygon cuts were cuts peculiar to edge intersection graph of point configurations in the plane. So the properties in this section are only for this case of intersection graphs.

Proposition 2. *Suppose $\{p_{i_1}, \ldots, p_{i_m}\} \subset S$ formed the vertices of a convex m-gon. Let $E_m = \{e_{i_s i_t} \in E : s - t \neq 0, \pm 1 (\mathrm{mod}\, m)\}$ be the diagonal edges of this m-gon. Let F be the convex hull of the incidence vectors of the independent sets including a set of diagonal edges of a triangulation of this m-gon. F forms a face of the independent set polytope P_{ind}. Its inequality is $\sum_{e_{i_s i_t} \in E_m} x_{i_s i_t} \leq m - 3$. The face F becomes a facet if and only if no edge in $E \setminus E_m$ intersects all sets of $m - 3$ non-intersecting edges in E_m.*

Corollary 3. *Let $S = \{p_1, \ldots, p_n\}$ be vertices of a convex n-gon. A face of the convex k-gon as in the proposition above forms a facet if and only if the k points are taken consecutively from the n-gon.*

We have shown that convex polygon cuts correspond to faces or facets of the independent set polytope P_{ind}. Since $P_{\max}$ was a face of P_{ind}, the inequalities of the faces of P_{ind} are valid for $P_{\max}$.

As already mentioned, P_{rel} is the relaxation of P_{ind} and $P_{\mathrm{rel}} \cap \{\boldsymbol{x} : \sum_{e_{ij} \in E} x_{ij} = M\}$ is the relaxation of $P_{\max}$. We started from solving Linear Programming on $P_{\mathrm{rel}} \cap \{\boldsymbol{x} : \sum_{e_{ij} \in E} x_{ij} = M\}$ and were looking for a minimum weight triangulation which is a vertex of $P_{\max}$.

The inequalities of faces of P_{ind} were fully used as cutting planes. Lifting of the non-facet faces of the above three types makes facets. Giving description to facets of other kind would be interesting.

6 MWT as maximization and Nemhauser-Trotter test

Another formulation for the minimum weight triangulation as the maximization problem, state as below, provides interesting insights to MWT.

$$\max \left\{ \sum_i (U - c_i)x_i \,\middle|\, x_i \in \{0,1\}, \quad x_i + x_j \leq 1 \; (\, i, j \text{ s.t. } e_i \text{ and } e_j \text{ cross}) \right\}$$

We use the value $\max(c_i) + \epsilon$ for U so that all the coefficients in the object function are positive. In MWT, the cardinality of the maximal independent set is fixed, so that the condition $\sum_i x_i = M$ can be dispensed with for positive costs. The problem then becomes a pure vertex packing problem and we can solve its LP relaxation efficiently by using the minimum-cost-flow algorithm.

Nemhauser and Trotter [13] showed that the variables with integer values in the relaxed LP problem of the original problem without cuts have the same value in the original vertex packing problem. This property means that we may gradually reduce the size of the problem by fixing the integer-valued variables, as by skeletons. Unfortunately, most of the variables have the value 1/2 in large cases, and we can solve only the small problems, say, at most 10 points, which were observed through large-scale computational experiments.

One approach to increase the number of variables with integer values would be to reduce the upper bound U, which results in negative coefficients for *long* edges. Edelsbrunner and Tan [8] showed that a triangulation which minimize the maximum edge length can be obtained in polynomial time. This means that the lower bound U_l for U in order to obtain a triangulation is available. Although the triangulation obtained with reduced U ($U_l < U < \max c_i$) is not necessarily the minimum weight triangulation, obviously we have the following:

Proposition 4. *If we obtain a triangulation by using* U ($U_l < U < \max c_i$)*, it is the minimum weight triangulation using edges shorter than* U*.*

It is open in what condition a triangulation obtained in this way is the minimum weight one, or how close to the minimum weight. But this approach may give a good, efficient and robust approximation. For example, in the example in Fig.6 for which neither β-skeleton nor LMT-skeleton work well, we obtain a triangulation which is actually the minimum weighted by using the upper bound $U = 2 \times \min c_i$.

7 Conclusion

We have presented a framework of applying the branch-and-cut paradigm to this problem, and have demonstrated its potential power by introducing new effective cuts, together with their theoretical analysis.

Triangulations in high dimensions are also of both theoretical and practical interest. Our branch-and-cut approach can be generalized by considering an intersection graph of simplices. This reversely elucidates the usefulness of investigating an intersection graph of triangles beside the intersection graph of all line segments in the planar case. Some of the authors have analyzed and making use of the structure of such intersection graphs in connection with enumerating triangulations [12, 14], and currently we are planning to computational investigate triangulations, or tetrahedralization, of points in the three-dimensional space.

Acknowledgment

The work by the second author was partially supported by the Grant-in-Aid of the Ministry of Education, Science, Sports and Culture of Japan.

References

1. D. Applegate, R. Bixby, V. Chvatal and B. Cook. Finding Cuts in the TSP (A preliminary report). *Technical Reports 95-05*, DIMACS, 1995.
2. M. Berkelaar. lp_solve. Available at `ftp://ftp.es.ele.tue.nl/pub/lp_solve`.
3. M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In *"Computing in Euclidean Geometry"* (2nd edition), Lecture Notes Series on Computing, Vol.4, World Scientific, 1995, pp.47–123.
4. S.-W. Cheng, M. J. Golin, and J. C. F. Tsang. Expected case analysis of β-skeletons with applications to the construction of minimum-weight triangulations. *Proc. 7th Canadian Conference of Computational Geometry*, 1995, pp.279–284.
5. S.-W. Cheng, N. Katoh and M. Sugai. A study of the *LMT*-skeleton. *Proc. 7th International Symposium on Algorithms and Computation*, Lecture Notes in Computer Science, Vol.1178, pp.256–265.
6. S.-W. Cheng and Y.-F. Xu. Approaching the largest β-Skeleton within a minimum weight triangulation. *Proc. 12th Annual ACM Symposium on Computational Geometry*, 1996, pp.196–203.
7. M. T. Dickerson and M. H. Montague. A (usually?) connected subgraph of the minimum weight triangulation. *Proc. 12th Annual ACM Symposium on Computational Geometry*, 1996, pp.204–213.
8. H. Edelsbrunner and T. S. Tan. A quadratic time algorithm for the minmax length triangulation. *SIAM J. Comput.*, 22:527–551, 1993.
9. J. M. Keil. Computing a subgraph of the minimum weight triangulation. *Computational Geometry Theory and Applications*, 4:13–26, 1994.
10. Y. Kyoda. A study of generating minimum weight triangulation within practical time. Master's Thesis, Department of Information Science, University of Tokyo, March 1996. Available at `http://naomi.is.s.u-tokyo.ac.jp/`
11. C. Levcopoulos and D. Krznaric. Quasigreedy triangulations approximating the minimum weight triangulation. *Proc. 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1996, pp.392–401.
12. T. Masada, H. Imai and K. Imai. Enumeration of regular triangulations. *Proc. 12th Annual ACM Symposium on Computational Geometry*, 1996, pp.224–233.
13. G. L. Nemhauser and L. E. Trotter. Vertex packings: structural properties and algorithms. *Mathematical Programming*, 8,2:232–248, 1975.
14. F. Takeuchi and H. Imai. Enumerating triangulations for products of two simplices and for arbitrary configurations of points. *Proc. 3rd International Computing and Combinatorics Computing Conference*, Lecture Notes in Computer Science, 1997.
15. C. A. Wang, F. Chin and Y.-F. Xu. A new subgraph of minimum weight triangulation. *Proc. 7th International Symposium on Algorithms and Computation*, Lecture Notes in Computer Science, Vol.1178, pp.266–274.