Triangulations Match Intersect Nicely

Oswin Aichholzer
Franz Aurenhammer
Michael Taschwer
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

Abstract

We prove that two different triangulations of the same planar point set always intersect in a systematic manner, concerning both their edges and their triangles. As a consequence, improved lower bounds on the weight of a triangulation are obtained by solving an assignment problem. The new bounds cover the previously known bounds and can be computed in polynomial time. As a by-product, an easy-to-recognize class of point sets is exhibited where the minimum-weight triangulation coincides with the greedy triangulation.

1 Introduction

The aim of this paper is to prove and discuss some surprising and rather general intersection properties of planar triangulations.

Given two triangulations of a point set, we can find a matching between their edge sets such that matched edges either cross or are identical. This theorem and a few related statements will be proved in Section 2.

The remaining part of the paper deals with applications of this result to the computation of minimum-weight triangulations. In Section 3 we identify special cases of point sets for which the minimum-weight triangulation can be computed efficiently, see Figure 2. Section 4 defines several lower bounds on the minimum-weight triangulation, and Section 5 describes some algorithms to compute these bounds. In Section 6 we will discuss possible applications of our results and some open questions.

2 The Matching Theorems

Let P be a set of n points in the plane. We assume that not all points lie on one line. Consider the set E of all line segments connecting two points of P and containing no other points of P. (If P is in general position then $|E| = \binom{n}{2}$.) The elements of E are called *edges*. Two distinct edges are said to cross iff they intersect in their interiors. In particular, two edges sharing only one endpoint are not considered as crossing. A *triangulation* T of P is a maximal set of non-crossing edges. It dissects the convex hull of P into triangular faces. We start by recalling a well-known fact on triangulations.

Lemma 1 *Every triangulation of P consists of $m = 3n - 3 - b$ edges, where b is the number of points in P which lie on the boundary of the convex hull of P. Every set of non-crossing edges consists of at most m edges.* □

The following is the main theorem of this paper.

Theorem 1 *Let P be a finite set of points in the plane and consider two triangulations R and B of P. There exists a perfect matching between R and B, with the property that matched edges either cross or are identical.*

Proof. For notational simplicity, let us color the edges in B as blue and the edges in R as red. We consider the intersection graph, G, of $R \cup B$. G represents each edge in R by a red node and each edge in B by a blue node. (Edges in $R \cap B$ are represented by two nodes.) Two nodes are connected

[0] *This paper is submitted to the 11th Annual Symposium on Computational Geometry, to be held in Vancouver, June 1994.*

by an arc if the corresponding edges either cross or are identical. Clearly, there are no arcs between nodes of the same color as edges in a single triangulation do not cross. Hence G is bipartite. Note that, by Lemma 1, the number of red nodes equals the number of blue nodes.

We prove that G contains a perfect matching by showing that G fulfills the Hall condition of the marriage theorem (see, e.g., [B]). This condition requires that, for each subset X of red nodes of G, the number of blue neighbors in G is at least $|X|$.

Let $R_1 \subseteq R$ and let B_1 collect each blue edge that either crosses or coincides with an edge in R_1. With this notation, the Hall condition reads $|B_1| \geq |R_1|$. Let B_2 be the complement of B_1 in B. We claim that $B_2 \cup R_1$ is a non-crossing set of edges. The edges of R_1 do not cross each other, since $R_1 \subseteq R$, and likewise, the edges of B_2 do not cross each other. An edge b of B_2 cannot cross an edge of R_1 because b would belong to B_1, otherwise. For the same reason, R_1 and B_2 are disjoint, which gives

$$|R_1| + |B_2| = |R_1 \cup B_2| \leq m = |B|,$$

by Lemma 1. Thus we get $|B_1| = |B| - |B_2| \geq |R_1|$, and the Hall condition is proved. □

It seems natural to try to prove Theorem 1 in a direct way, without resorting to the marriage theorem. For example, any triangulation of a planar point set can be changed into the Delaunay triangulation (and thus into an arbitrary other one) by repeated application of so-called edge flips, see [Law]. An edge flip exchanges the diagonals of a convex quadrilateral in the current triangulation and thus naturally corresponds to a match of the involved edges. We did not succeed in a proof of Theorem 1 based on the flipping paradigm, however.

Corollary 1 *Let P be a finite set of points in the plane. Let R be a set of non-crossing edges between points of P and let T be a triangulation of P. Then there is a matching between the edges in R and some edges in T (an injective mapping from R to T) such that every edge of R is either matched with the identical edge in T or with an edge which crosses it.*

Proof. Since R can be extended to a triangulation the corollary follows from Theorem 1. □

Our proof mainly exploits the property of triangulations expressed in Lemma 1. A version of Lemma 1 concerning triangles instead of edges is obvious.

Lemma 2 *Every triangulation of P dissects the plane into $2n-2-b$ interior triangular faces plus the exterior face, where b is the number of points in P which lie on the boundary of the convex hull of P. Every set of non-crossing edges dissects the plane into at most $2n-1-b$ connected components.*

This enables us to prove, with exactly the same technique as above, the next theorem on triangles.

Theorem 2 *Let P be a finite set of points in the plane and consider two triangulations R and B of P. There exists a perfect matching between the set of triangles of R and the set of triangles of B, with the property that matched triangles either overlap or are identical.*

Figure 1 displays a perfect triangle matching for two triangulations of a convex polygon.

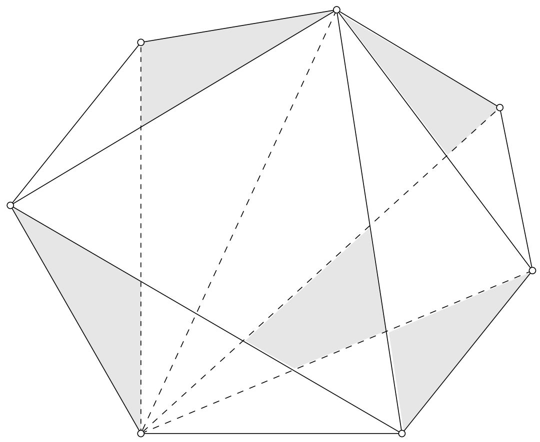

Figure 1: Perfect matching of overlapping triangles between a "solid" and a "broken" triangulation of a 7-gon. Every shaded area is the intersection between two matched triangles.

Our matching theorems have been found independently by Cheng and Xu [CX]. They generalized the result to the framework of arbitrary independence systems.

An *independence system* $\mathcal{I}$ is a non-empty collection of subsets of a (finite) ground set E which is closed under taking subsets: If $A \in \mathcal{I}$ and $B \subset A$ then $B \in \mathcal{I}$. The elements of $\mathcal{I}$ are called the *independent* sets, the remaining subsets of E are called *dependent*. A *circuit* of $\mathcal{I}$ is a minimal dependent set.

In our example of triangulations, a set of non-crossing edges (or of non-overlapping triangles) may be considered independent. The circuits of this independence system have two elements; they are the pairs of crossing edges (or of overlapping triangles, respectively).

Theorem 3 *Let $R \in \mathcal{I}$ be any independent set, and let $B \in \mathcal{I}$ be an independent set of maximum cardinality in $\mathcal{I}$. Then there is an injective mapping $g: R \to B$ such that for every edge $e \in R$ we have $g(e) = e$, or $\{g(e), e\}$ is contained in a circuit.*

The proof shows that the Hall condition is fulfilled, following the lines of the proof of Theorem 1. Theorems 1 and 2 are corollaries of Theorem 3.

Cheng and Xu [CX] also strengthened Theorem 2 by requiring in addition that matched triangles have at least one vertex in common. The proof shows the Hall condition by an argument about sums of angles.

Let us now briefly take the algorithmic standpoint and ask how expensive it is to compute the perfect matchings whose existence is predicted in Theorems 1 and 2. We restrict attention to edge matchings; the situation for triangles is similar. As before, assume we are given a red triangulation R and a blue triangulation B of a set P of n points. The first approach that comes into mind is to use a standard bipartite matching algorithm for the intersection graph defined by R and B.

In a first pass, edges showing up in both triangulations are eliminated since their matches are determined uniquely. In particular, edges bounding the convex hull of P are those. The intersection graph, G, of the remaining edges consists of $O(n)$ nodes and $k = O(n^2)$ arcs, where k is the total number of crossings between R and B. We compute a perfect matching in G by applying the bipartite matching algorithm of Hopcroft and Karp [HK]; see also [T]. A direct application would result in $O(k\sqrt{n})$ time and $O(k)$ space. We can reduce the space requirement to $O(n)$ by extracting the adjacency lists of G's nodes on-line. In an $O(n)$-time preprocessing step we locate, for each point $p \in P$, the incident red edges among incident blue triangles by merging two sorted lists of angles. To obtain the neighbors of, say, a red node we simply proceed along the red edge it represents, as long as this edge intersects blue triangles.

This result is quite satisfactory for situations where the number k of crossings is not substantially larger than the size n of the underlying set of points. If $k = \Theta(n^2)$, a $\Theta(n^2\sqrt{n})$ time algorithm is obtained. As both the input (two triangulations) and the output (the matched edge pairs) are of size $O(n)$, a solution that exploits the geometric nature of the problem in a better way is desirable.

The idea of constructing a perfect matching in a greedy manner suggests itself: After matching identical edges, we match each unmatched red edge with the first yet unmatched blue edge crossing it which we find. From the theory of bipartite graphs it is known that the greedy method always matches at least half of the nodes of the graph, provided it contains a perfect matching. Hence at least half of the matches among crossing edges will be found. However, the illusion that geometry would be strong enough to enforce a perfect matching is destroyed by an easy example.

By observing that the (at least three) convex hull edges are matched by the greedy method anyway, we get:

Lemma 3 *Any greedy matching leaves strictly less than half of the edges unmatched.*

We do not further pursue the issue of finding a perfect matching here. We rather concentrate on applications of the edge matching theorem to minimum-weight triangulations which we consider —apart from the matching theorems themselves— as the most interesting part of our paper.

3 Light Triangulations

The intersection properties of planar triangulations expressed in the foregoing section should have various applications in combinatorial and computational geometry. In the present paper, two applications of Theorem 1 (perfect matching between edges) to minimum-weight triangulations are demonstrated.

In this section, we exhibit a class of planar point sets where the minimum-weight triangulation coincides with the greedy triangulation and thus can be computed easily and in polynomial time. An efficient algorithm for recognizing such point sets is given in Section 5.

As before, let P be a set of n points in the plane, and let E be the set of all edges defined by P. The length (weight) of an edge $e = pq \in E$ is just the Euclidean distance between p and q and

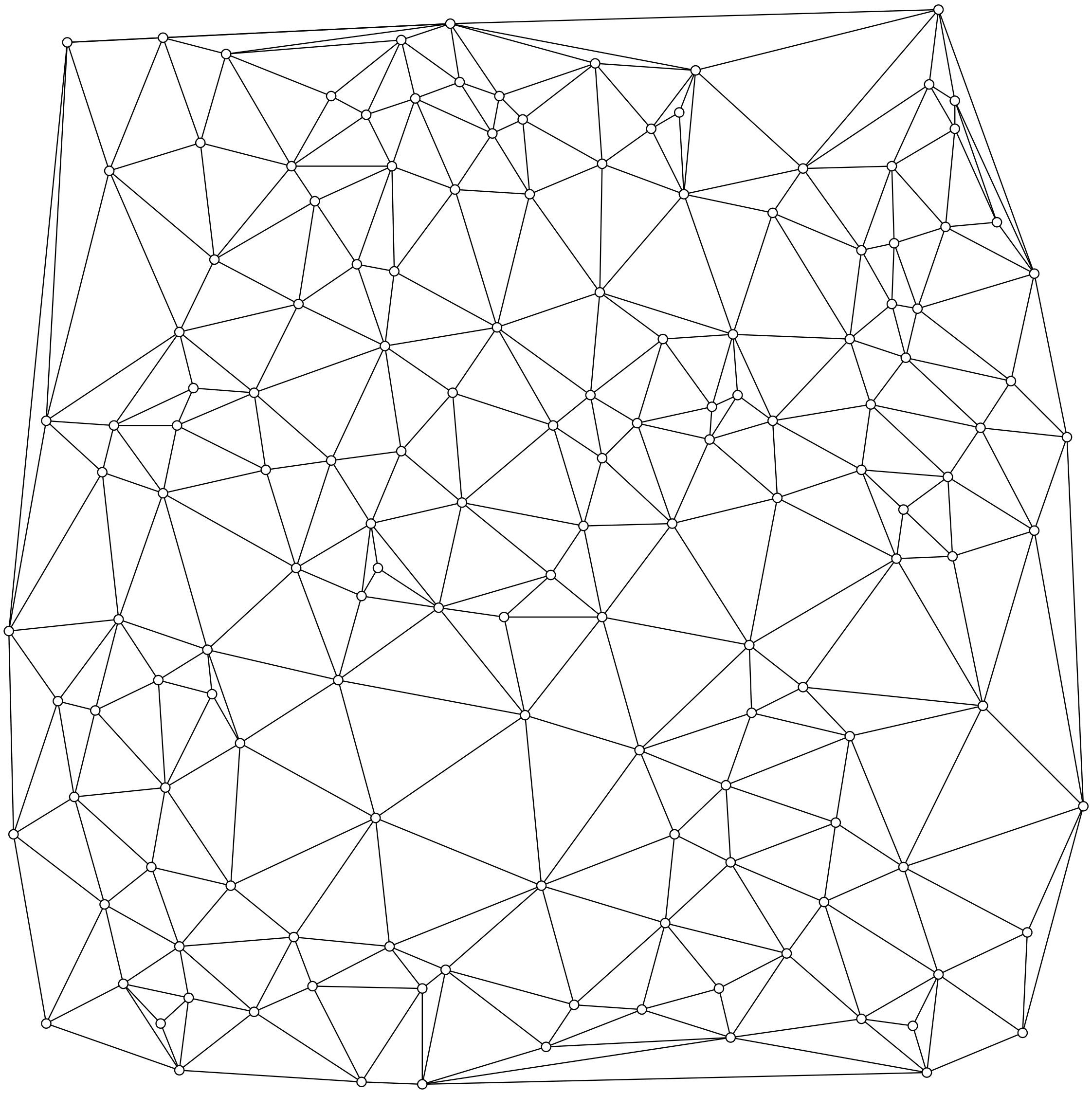

Figure 2: Minimum weight (light) triangulation for 150 points.

it will be denoted by $|e|$ or $|pq|$. To keep things simple, we will henceforth assume general position so that all $\binom{n}{2}$ distances between points of P are distinct. (Otherwise, if a coincidence of two edges with the same length happens, we may for example give preference to the edge which comes first in some given prespecified numbering.) The weight $w(R)$ of a set R of edges is the sum of the lengths of its edges.

The *minimum-weight triangulation* T^* is defined to have $w(T^*) \leq w(T)$ for all triangulations T of P. Computing a minimum-weight triangulation is an important and interesting problem, whose complexity status is unknown. It is one of the open problems listed at the end of Garey and Johnson's book about NP-completeness [GJ].

We first reconsider a type of triangulations that has already been looked at from various points of view. (We refer the reader to [Lam] for a comprehensive survey on triangulations.) A *locally short triangulation* of the point set P is one which includes, for each convex quadrilateral Q it contains, the shorter diagonal of Q. Five points are enough to see that the locally short triangulation is not unique. Any given triangulation of P can be made locally short by finitely many diagonal flips, as each flip decreases the total edge length. Unfortunately, a locally short triangulation fails to fulfill length optimization criteria. It is well known that the flipping procedure can get caught in a local optimum: It need not yield the minimum-weight triangulation. On the other hand note that a minimum-weight triangulation must of course be locally short. Our interest here is in a subclass of locally short triangulations. Let us call an edge $e \in E$ *light* if there is no edge in E that crosses e and is shorter than e. Light edges obviously do not cross (under the above general position assumption), so the set L of light edges can form at most a triangulation of P. If L actually is a triangulation then it has to be unique and, of course, locally short. We then call L the *light triangulation* of P. See Figure 2 for an example. (In fact, it is quite easy to draw a light triangulation by 'well' distributing the points).

In the light of Theorem 1, it is easy to prove length optimality.

Theorem 4 *If a planar point set P admits a light triangulation then this is the minimum-weight triangulation for P.*

Proof. Let T be any triangulation of P. We show $w(T) \geq w(L)$, for L being the light triangulation of P. Consider a perfect matching as in Theorem 1 between the edges of T and the edges of L. For each matched pair of edges $e \in T$ and $e' \in L$, either $e = e'$ or e crosses e', in which case we know that $|e| > |e'|$ by the lightness of L. Summing over all edges gives $w(T) \geq w(L)$. □

A popular kind of locally short triangulations is the *greedy triangulation*. This structure is defined procedurally. It is obtained by iteratively inserting the shortest edge of E that does not cross previously inserted edges. While, in general, a locally short triangulation need not contain all light edges (Figure 3), the greedy triangulation does so. A light edge e never can be blocked by previously inserted edges as E does not contain any shorter edge crossing e. So, if a light triangulation exists for P, it can be found quickly by applying one of the various known greedy triangulation algorithms. More importantly, by Theorem 4 the minimum-weight triangulation of P can be computed in polynomial time in this case, in fact in $O(n^2)$ time and $O(n)$ space, see Levcopoulos and Lingas [LL] and Wang [W]. See also Goldman [G] for an easy-to-implement $O(n^2 \log n)$ time variant, and [DDMW, DRA] for fast expected-time algorithms.

Selecting all light edges from a given greedy triangulation takes polynomial time, naively $O(n^3)$. We will improve this bound to $O(n^2 \log n)$ in Section 5.

It is noteworthy that, in general, not all light edges occur in a minimum-weight triangulation T^*.

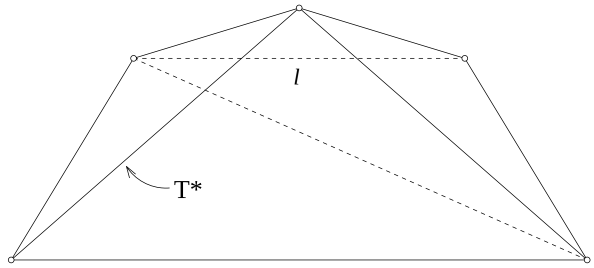

Figure 3: Light edge l not included in the minimum-weight triangulation T^*.

Figure 3 gives an example. However, recent results [K, YXY] imply that certain subsets of light edges always have to be there. Keil [K] shows that T^* always includes the so-called $\sqrt{2}$-skeleton of P. This structure consists of all edges pq such that the two circles through both p and q and with diameter $\sqrt{2}\,|pq|$ are empty of points in P; see Figure 4. It

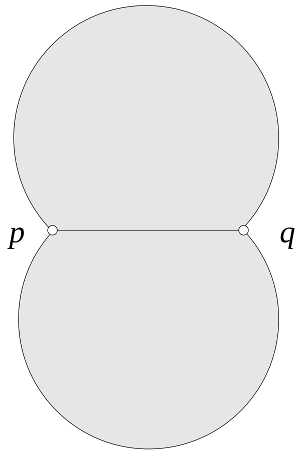

Figure 4: Keil's empty region defining the $\sqrt{2}$-skeleton.

is not difficult to verify that this condition forces $|pq|$ to be light. The $\sqrt{2}$-skeleton is a subset of the Gabriel graph (1-skeleton) which itself is known to be not part of the minimum-weight triangulation.

Yang et al. [YXY] prove that a condition stronger than lightness implies inclusion of pq in T^*: For any edge xy crossing pq, $|pq| \leq \min\{|px|, |py|, |qx|, |qy|\}$. By the quadrilateral inequality, xy then has to be the longest of the six edges involved. Hence pq is a light edge, and we obtain the region in Figure 5 whose emptiness implies $pq \in T^*$.

4 Lower Bounds for $w(T^*)$

In this section, we offer new methods for computing lower bounds for the weight of planar triangula-

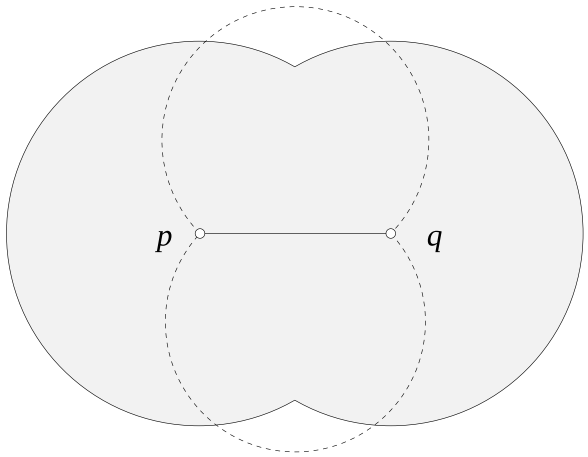

Figure 5: Empty region of Yang et al.

tions. Apart from the trivial lower bound (the sum of the $|T^*|$ shortest edges in E), all bounds known to us require the knowledge of a set of edges which is a subset of T^*. Though there is recent insight into this problem [K, YXY], the quality of such bounds usually suffers from the small cardinality of such subsets. Theorem 1 allows us to prove a lower bound for $w(T^*)$ in a completely different way by solving an assignment problem.

Theorem 5 *Let R be a noncrossing set of edges and let T^* be the minimum-weight triangulation. Let $X(R,E)$ denote the set of all injective mappings (matchings) $g\colon R \to E$ with the properties required in Theorem 1: $g(e) = e$ or $g(e)$ crosses e. Then we have the following lower bound.*

$$w(T^*) \geq \min_{g \in X(R,E)} \sum_{e \in R} |g(e)|. \tag{1}$$

Proof. Let $g^* \in X(R,E)$ be a matching $g^*\colon R \to T^*$ whose existence is guaranteed by Theorem 1 and Corollary 1. We have

$$w(T^*) = \sum_{e \in T^*} |e| \geq \sum_{e \in R} |g^*(e)| \geq \min_{g \in X(R,E)} \sum_{e \in R} |g(e)|.$$

□

The set $X(R,E)$ is just the set of complete matchings in a bipartite graph, and hence we can optimize over this set in polynomial time. Note that edges in the set $g(R) = \{\, g(e) \mid e \in R \,\}$ may cross each other. Solving the assignment problem with the constraint that $g(R)$ is non-crossing yields a solution to the original minimum-weight triangulation problem. (However, in its general form the assignment problem can be shown to be NP-complete by reduction from independent set [GJ].)

Stated the other way, when R is a triangulation and $g_{opt}(R)$ happens to be non-crossing for the optimal assignment g_{opt} in (1) then $g_{opt}(R)$ must be minimum-weight.

Corollary 2 *Let R be a triangulation and let $g_{opt} \in X(R,E)$ satisfy*

$$\sum_{e \in R} |g_{opt}(e)| = \min_{g \in X(R,E)} \sum_{e \in R} |g(e)|.$$

If the set $g_{opt}(R)$ is non-crossing then it is a minimum-weight triangulation.

The optimal assignment g_{opt} has a further property.

Theorem 6 *g_{opt} matches no edge e to a longer edge, that is, $|g_{opt}(e)| \leq |e|$ for all $e \in R$.*

Proof. Suppose that a matching $g \in X(R,E)$ has $|g(e)| > |e|$ for some edge e. Then we can improve g by assigning e to itself, i. e., by setting $g(e) = e$ and leaving g unchanged for the remaining edges. Since no other edge $f \in R$ crosses e we still obtain a matching, and hence the original matching g cannot be optimal. □

If we relax the requirements on e in Theorem 5 and permit all injective mappings from R to E we get the trivial lower bound: the sum of the $|R|$ shortest edges in E. This shows that our bound can never be worse than the trivial one.

We may weaken the bound of Theorem 5 in a different way, by neglecting the fact that the matching g must assign different edges of R to different edges of T^*. The resulting bound is very easy to compute once some auxiliary information on the edges in R has been precomputed. For an edge $e \in E$, let us define its *excess* $\varepsilon(e)$ as

$$\varepsilon(e) = \max\{0,\ |e| - \lambda(e)\},$$

where $\lambda(e)$ is the length of the shortest edge crossing e. If e is crossed by no edge (for example, if e is an edge of the convex hull) then we set $\lambda(e) = \infty$, hence $\varepsilon(e) = 0$. Edges which have no crossing edges are called *unavoidable* edges; they belong to every triangulation, see Xu [X]. An edge has no shorter edge which crosses it iff its excess vanishes. We have called such edges light in the preceding section.

Theorem 7 *Let R be a noncrossing set of edges and let T^* be a minimum-weight triangulation. Then*

$$w(T^*) \geq w(R) - \sum_{e \in R} \varepsilon(e).$$

Proof. Let $g^*: R \to T^*$ be a matching as in the proof of Theorem 5. Then for all $e \in R$, either $g^*(e)$ crosses e or $g^*(e) = e$. In the first case, we have

$$|e| - \varepsilon(e) \leq |g^*(e)| \tag{2}$$

by the definition of the excess $\varepsilon(e)$. In the second case, the same inequality holds because $\varepsilon(e) \geq 0$. By summing (2) over all $e \in R$ we get

$$w(R) - \sum_{e \in R} \varepsilon(e) \leq \sum_{e \in R} |g^*(e)| \leq \sum_{e \in T^*} |e| = w(T^*).$$

□

The consequences of this theorem for the set $L \subset E$ of light edges are particularly interesting. Since L is a noncrossing set and since $\varepsilon(e) = 0$ for all edges $e \in L$, we have:

Corollary 3

$$w(T^*) \geq w(L).$$

We stress the fact that, despite $L \not\subseteq T^*$ in general, summing up the light edges provides a valid lower bound. This bound covers the bounds in [K, YXY] as the subsets of T^* considered there are built of light edges only; see Section 2. Generally, if $R' \subset R$, then the bound given by R in Theorem 7 is stronger than the bound given by R'. Thus it pays off to complete the non-crossing set R (or L) to a triangulation before applying Theorem 7. Note finally that if L happens to be a triangulation then $w(L) = w(T^*)$ and we obtain Theorem 4 of Section 2.

5 Algorithmic Issues

We now address the algorithmic issues raised in Sections 3 and 4.

5.1 Computing excesses and light edges

Theorem 4 asks for a method to decide whether a given planar n-point set P admits a light triangulation, or in other words, whether the light edges triangulate P. Also, to compute the lower bound in Corollary 3 all light edges realized by P are required. The greedy triangulation contains all light edges. So we only need to test greedy edges for lightness. More generally, the bound in Theorem 7 needs knowledge of the excess of all edges in a given non-crossing set.

We solve these problems in time $O(n^2 \log n)$ and space $O(n)$ by giving an algorithm for the following problem: Given an arbitrary triangulation T of P, compute $\lambda(e)$ for each $e \in T$, where $\lambda(e)$ is the length of the shortest edge crossing e. As a by-product, all unavoidable edges spanned by P are detected as those getting $\lambda(e)$ never assigned.

We describe an $O(n \log n)$-time routine which, for a given point $p \in P$, computes

$$\lambda_p(e) = \min_{q \in P} \{\, |pq| \mid pq \text{ crosses } e \,\}$$

for all $e \in T$. Calling the routine for all $p \in P$ and maintaining the minimum for each e then gives $\lambda(e)$.

Given the point p, we set up a semi-dynamic data structure for a point set Q and for the following type of queries: For a wedge V with apex p and angle less than π, return the closest point to p in $Q \cap V$. We initialize Q with P and allow only deletions from Q. The data structure we use is a binary search tree whose leaves store Q in cyclic order around p and whose interior nodes store minimum distances from p to all points in the subtree.

We then query the structure with wedges spanned by edges e in the triangulation T. The queries and deletions are carried out in a specific order. This guarantees that, when we query for the wedge belonging to edge e, Q contains no point that lies between e and p but includes all points that lie opposite to p with respect to e.

To this end, a topological order for the edges of T with respect to the in-front/behind relation as seen from p is necessary. Problems may arise if this relation contains cycles. (The class of regular triangulations, and thus the Delaunay triangulation, is acyclic in this sense [DFPN, E], but a general triangulation is not.) We circumvent this difficulty by cutting T with a ray emanating from p, and re-triangulating where necessary, in $O(n)$ time. For an edge e cut into e' and e'' we clearly have $\lambda_p(e) = \min\{\lambda_p(e'), \lambda_p(e'')\}$.

A topological order now is obtained in $O(n)$ time by starting with the set of triangles incident to p, and adding adjacent triangles one by one while keeping the invariant that the boundary of their union U is star-shaped as seen from p. At each point in time, and the search tree stores exactly the points of P exterior to U. As queries and deletions take $O(\log n)$ time each, the claimed $O(n \log n)$ time bound is obtained.

Theorem 8 *Let P be a set of n points in the plane.*

(i) Given a triangulation T of P, the excesses for all edges in T can be computed in $O(n^2 \log n)$ time and $O(n)$ space.

(ii) The excesses for all edges spanned by P can be computed in $O(n^3 \log n)$ time and $O(n^2)$ space.

(iii) The set of light edges spanned by P can be computed in time $O(n^2 \log n)$ time and $O(n)$ space.

Proof. Statement (i) follows from the foregoing discussion. Statement (ii) follows from (i) since we can cover all edges by n triangulations: We simply connect a fixed point p to all other points and complete this to a triangulation. Repeating this for all points p, we will have covered every edge between two points of P at least once. For proving (iii) it is sufficient to note that the greedy triangulation of P can be computed in the claimed time and space bounds. □

5.2 Computing the strong optimal assignment bound

The problem whose solution is required for applying Theorem 5 is an assignment problem between two edge sets R and E, where R is a given triangulation and E is the set of all edges between points of P. (In fact R could be any non-crossing set but we restrict our attention to a triangulation.) The straightforward application of the standard algorithm for the assignment problem leads to a complexity of $O(n^4)$ time and $O(n^3)$ space.

Note however that the weights are in fact not associated with the *arcs* of the bipartite graph in which we find the matching but with the *node* set E. The subsets of nodes in a graph which are matchable (for which there exists a matching which covers all of them) is a matroid, in our case of a bipartite graph a transversal matroid, see [La, Corollary 7.4.3, p. 272, and Section 5.4, p. 192]. (This is in contrast to the situation for the arc sets forming a matching.) It follows from basic matroid theory (see [La, T]) that it is possible to solve the problem by a greedy algorithm on the set E: We start with the empty matching and process the edges of E in order of increasing weight. For each edge we try to augment the current matching to include the corresponding node. A node which is matched in this process will never become unmatched. We may stop if we have $|R|$ matching pairs.

Let us focus on the form of an augmenting path which we are looking for when processing an edge e_0. The path goes from e_0 to an adjacent node r_1 (an edge of R), from r_1 to the node e_1 to which r_1 is matched, from e_1 to an adjacent node r_2, from r_2 to the node e_2 to which r_2 is matched, and so on until it terminates in an unmatched node $r_i \in R$, $i \geq 1$. The augmentation consists of exchanging the $i-1$ matching arcs with the i non-matching arcs on this path.

We may search for such an augmenting path starting from e_0 using any graph search method, for example breadth-first search. We make several observations:

(i) We only need to explore the list of adjacent nodes for the edges of E; when we are at a node of R we simply proceed to the node of E to which it is matched. Since the neighbors of $e \in E$ are the edges of the triangulation R which cross e, we can find these neighbors by walking through the triangulation, without any need to store adjacency lists.

(ii) If a node of E cannot be matched we can ignore it in the future. We need not remember to which nodes it is adjacent. Thus if $e \in E$ is not matched in the optimal solution, its incident arcs are explored only once during the whole algorithm.

(iii) If we have processed a node in R or E during an unsuccessful search for an augmenting path we know that no unmatched node of R can be reached from this node, and we may skip the search from this node in all subsequent searches. This situation remains unchanged as long as no augmentation occurs. Thus, between two successful augmentations, every edge of the *current graph* is visited at most once. The current graph is the graph spanned by the nodes R and the subset E' of nodes of E which are currently matched.

(iv) There are $|R| = O(n)$ successful augmentation steps. Hence, if $E_{opt} = \{ g_{opt}(e) \mid e \in R \}$ denotes the set of matched nodes of E in the optimal matching g_{opt}, each of the arcs between R and E_{opt} is visited at most $O(n)$ times.

The only operation which we have not discussed is the generation of all segments in E according to length. Dickerson et al [DDS] have shown that the c shortest edges in an n-point set P may be generated in $O((c+n)\log n)$ time and $O(c+n)$ space. By trying successively the values $c = n, 2n, 4n, 8n, \ldots$ we can make sure that we never generate more than twice as many edges as we actually need. The $O(\log n)$ overhead for each of the c edges generated is also sufficient to locate the first triangle in the triangulation R which is cut by that edge. Summarizing, we have the following theorem.

Theorem 9 *Let $E_{opt} = \{\, g_{opt}(e) \mid e \in R \,\}$ denote the set of edges to which the edges of R are assigned in the optimal matching g_{opt} of (1) in Theorem 5. Let $K = O(n^2)$ denote the number of crossings between E_{opt} and R.*

Furthermore let $E_{\text{small}} \subseteq E$ be the set of edges which are not longer than the longest edge of E_{opt}, and let $L \leq |E_{\text{small}}| \cdot |R| = O(n^3)$ denote the number of crossings between E_{small} and R.

Then the optimal matching g_{opt} can be computed in $O(Kn + L + |E_{\text{small}}| \cdot \log n) = O((K + |E_{\text{small}}|)n) = O(n^3)$ time and $O(n + |E_{\text{small}}|) = O(n^2)$ space.

Since we do not know the largest edge of E_{opt} beforehand we may take, by Theorem 6 the largest edge weight of R in the definition of E_{small} in order to get an a-priori upper estimation. In either case, unavoidable edges such as the boundary edges, which belong to every triangulation, can be ignored when computing the longest edge of E_{opt} or R.

The worst-case time bound of $O(n^3)$ is rather high, but the explicit parameters of the complexity indicate that for a good starting triangulation R the complexity might be quite good. (Within the worst-case time of $O(n^3)$, the edges can also be generated according to length in $O(n)$ space by a more primitive method.)

Theorem 6 also implies that we may skip every arc between $e \in E$ and a node $r \in R$ which is shorter than e. However, we can make use of this only at the expense of storing the current graph.

6 Conclusion and Open Problems

We are planning to use our bounds in a branch-and-bound algorithm to compute the minimum-weight triangulation for arbitrary point sets. The bounds of Theorems 5 and 7 can be strengthened for subproblems where some specified edges are forced into the solution or excluded from the solution.

We hope that a practically efficient algorithm for computing the minimum-weight triangulation may make experiments possible which lead to a better understanding of the properties of minimum-weight triangulations, with the ultimate goal of resolving the complexity status of the problem.

Various open questions are raised by the results in the present paper. Below is a list of what seemed most interesting to us.

The point sets for which the light edges form a triangulation are interesting. What good properties do they and their associated triangulations have?

Given a planar n-point set P, can it always be extended by adding, say, $O(n)$ points so that it admits a light triangulation? Can we find such "enlightening" points in polynomial time? These questions might be interesting from the point of view of engineering applications, where only some points on the boundary rather than the complete set of points of the triangulation are fixed in advance.

Can we bound the quality of the bounds of Theorems 5 and 7 when R is, for instance, the greedy triangulation or the Delaunay triangulation? (What is the maximum ratio between the two sides of the inequality?)

Theorems 1 and 2 are easily seen to hold for triangulations of arbitrary polygonal regions, possibly with holes. The latter theorem (perfect matching between overlapping triangles) defines a mapping of a polygonal region onto itself. Do such mappings have applications to geometric searching problems? What properties do such mappings have in the limit, i.e., when the number of points approaches infinity?

A triangulation can be viewed either as a set of edges or as a set of triangles. When we take the weight of a triangle to be its perimeter, the minimum-weight triangulation problem is identical for both formulations. (In the triangle formulation, all edges are counted twice except for the boundary edges, which are counted once but are of constant length.) Nevertheless, we get two different greedy algorithms from the two formulations. We are not aware of treatments of the "triangle-greedy" algorithm so far. Since there are $O(n^3)$ triangles but only $O(n^2)$ edges it seems plausible that it should beat the usual "edge-greedy" algorithm in practice. (We have examples which show that none of the two greedy algorithms beats the other for all problems.) How fast can the triangle-greedy triangulation be computed?

Acknowledgements. We thank Xu Yin-Feng for sharing his ideas in enlightening discussions. Thanks also go to Scot Drysdale, Herbert Edelsbrunner, and Gerhard Wöginger for discussions on the presented topic.

The Research of Oswin Aichholzer was supported by the Austrian Ministry of Science and the Jubiläumsfond der Österreichischen Nationalbank.

References

[B] B. Bollobás, *Graph Theory. An Introductory Course*, Springer Verlag, 1979.

[CX] S.-W. Cheng and Y.-F. Xu, *Constrained independence system and triangulations of planar point sets*, Manuscript, Dept. of Comput. Sci., The Hong Kong University of Science & Technology, 1994.

[DFPN] L. de Floriani, B. Falcidieno, C. Pienovi, and G. Nagy, *On sorting triangles in a Delaunay tessellation*, Report, Ist. Mat. Appl., Consiglio Nazionale delle Ricerche, Genova, Italy, 1988.

[DDMW] M. Dickerson, R. L. Drysdale, S. McElfresh, and E. Welzl, *Fast greedy triangulation algorithms*, Proc. 10th Ann. Symp. Computational Geometry (1994), 211–220.

[DDS] M. Dickerson, R. L. Drysdale, and J.-R. Sack, *Simple algorithms for enumerating interpoint distances and finding k nearest neighbors*, Int. J. Computational Geometry & Appl. 3 (1992), 221-239.

[DRA] R. L. Drysdale, G. Rote, O. Aichholzer, *A simple linear time greedy triangulation algorithm for uniformly distributed points*, Manuscript, 1994.

[E] H. Edelsbrunner, *An acyclicity theorem for cell complexes in d dimensions*, Combinatorica **10** (1990), 252–260.

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

[G] S. Goldman, *A space efficient greedy triangulation algorithm*, Inf. Process. Lett. **31** (1989), 191–196.

[HK] J. E. Hopcroft and R. Karp, *An $n^{\frac{5}{2}}$ algorithm for maximum matchings in bipartite graphs*, SIAM J. Comput. **2** (1973), 225–231.

[K] M. Keil, *Computing a subgraph of the minimum weight triangulation*, Computational Geometry: Theory and Applications **4** (1994), 13–26.

[LL] C. Levcopoulos and A. Lingas, *Fast algorithms for greedy triangulation*, BIT **32** (1992), 280–296.

[Lam] T. Lambert, *Empty-shape triangulation algorithms*, Ph. D. Thesis, Dept. of Comput. Sci., Univ. of Manitoba, 1993.

[La] E. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart, and Winston, New York, 1976.

[Law] C. L. Lawson, *Software for C^1 surface interpolation*, In: Mathematical Software III, J. Rice (ed.), Academic Press, New York, 1977.

[T] R. E. Tarjan, *Data Structures and Network Algorithms*, SIAM Press, Philadelphia 1987.

[W] C. A. Wang, *Efficiently updating the constrained Delaunay triangulations*, BIT **33** (1993), 238–252.

[X] Y. Xu, Minimum weight triangulation problem of a planar point set, Ph. D. thesis, Institute of Applied Mathematics, Academia Sinica, Beijing, 1992.

[YXY] B.-T. Yang, Y.-F. Xu, and Z.-Y. You, *A chain decomposition algorithm for the proof of a property on minimum weight triangulations*, In: Algorithms and Computation, Proc. Conf., Beijing 1994, Lecture Notes in Computer Science 834, Springer-Verlag, 1994, pp. 423–427.