Enumerating pseudo-triangulations in the plane

Sergey Bereg*

Abstract

A pseudo-triangle is a simple polygon with exactly three convex vertices. A pseudo-triangulation of a finite point set S in the plane is a partition of the convex hull of S into interior disjoint pseudo-triangles whose vertices are points of S. A pointed pseudo-triangulation is one which has the least number of pseudo-triangles. We study the graph G whose vertices represent the pointed pseudo-triangulations and whose edges represent flips. We present an algorithm for enumerating pointed pseudo-triangulations in $O(\log n)$ time per pseudo-triangulation.

Keywords: pseudo-triangulation, flip, enumeration, reverse search.

1 Introduction

Pseudo-triangles and pseudo-triangulations have received much attention recently because of applications in visibility [15, 16], ray shooting [15], collision detection [13], and rigid motion [7, 20]. There are many open questions related to pseudo-triangulations [21].

A pseudo-triangle is a simple polygon with exactly three convex vertices. For a set S of n points in the plane, a *pseudo-triangulation* T is defined as partition of the convex hull of S into interior disjoint pseudo-triangulations whose vertices are points of S (each point of S is a vertex of T and vice versa). A *minimum* or *pointed pseudo-triangulation* introduced by Streinu [20] is a pseudo-triangulation with the least number of edges among all pseudo-triangulations of S. Streinu [20] showed that any pointed pseudo-triangulation has $2n-3$ edges. Equivalently, we can define a pointed pseudo-triangulation as a pseudo-triangulation with minimum number of pseudo-triangles. Any pointed pseudo-triangulation has $n-2$ pseudo-triangles since $n_f = n_e - n + 1 = n - 2$ by Euler's formula.

Recent results on pseudo-triangulations include [1, 2, 12, 18]. Randall *et al.* [18] gave closed form expressions for the number of triangulations and the number of pointed pseudo-triangulations when S has only one point inside its convex hull. For a point set in general position, they proved an upper bound for the number of pseudo-triangulations. Aichholzer *et al.* [1] investigated the number of pseudo-triangulations generated by n points in the plane. They proved that the least number of pseudo-triangulations is attained when points are in convex position.

We obtained an algorithm [4] for enumeration of triangulations in $O(\log \log n)$ time per triangulation. The algorithm is based on *flips*, small modifications of a triangulation. Pseudo-triangulations admit flips as well, see Fig. 1. It allows us to define, for a point set S, a *graph of pseudo-triangulations* whose vertices are the pointed pseudo-triangulations of S and whose edges are the flips of interior pseudo-triangulation edges. Recently, Brönnimann *et al.* [6] described an

*Department of Computer Science, University of Texas at Dallas, Box 830688, Richardson, TX 75083, USA. E-mail: besp@utdallas.edu

algorithm for enumerating all pointed pseudo-triangulations using an efficient technique of Pocchiola and Vegter [17] for finding a flip in $O(1)$ amortized time. Unfortunately the complexity of the algorithm is unknown[1] but the required space is quadratic. It follows from the algorithm that the graph of pseudo-triangulations is connected.

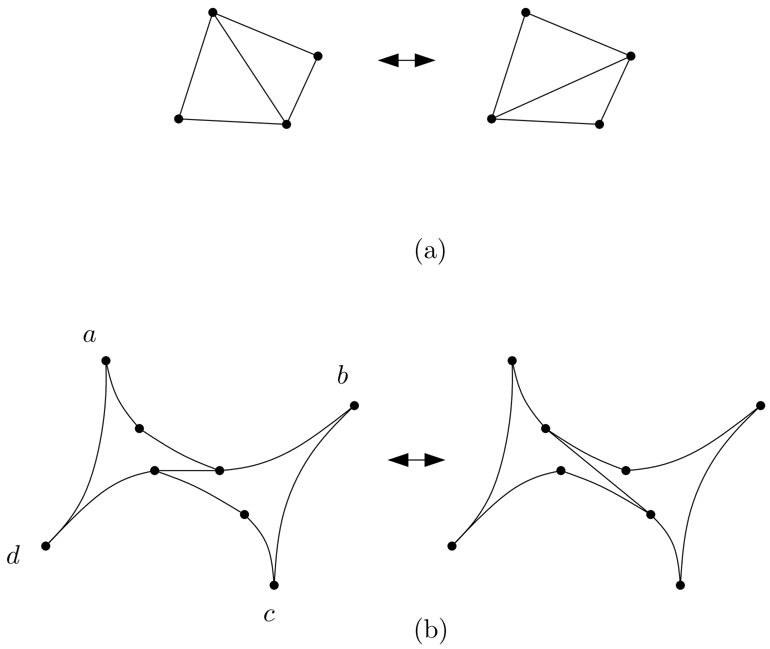

Figure 1: (a) Flip of triangles. (b) Flip of pseudo-triangles.

Very recently, Rote *et al.* [19] introduced a polytope of pointed pseudo-triangulations of a point set in the plane, defined as the polytope of infinitesimal expansive motions of the points subject to certain constraints on the increase of their distances. The polytope possesses useful properties and its 1-skeleton is the graph of pseudo-triangulations.

We present an algorithm for enumerating all pointed pseudo-triangulations in $O(\log n)$ time per pseudo-triangulation using linear space. The algorithm is based on reverse search technique by Avis and Fukuda [3]. As by-product it implies that the graph of pointed pseudo-triangulations is connected. Our algorithm can be used to list the vertices of the polytope of pointed pseudo-triangulations [19]. It also generates a spanning tree of the graph of pseudo-triangulations.

Aichholzer *et al.* [2] considered the problem of counting the number of pointed pseudo-triangulations of a point set. They showed that every pseudo-triangulation contains a *zigzag path*. It is used to count pointed pseudo-triangulations efficiently by enumerating zigzag paths in $O(n^2)$ time per path.

2 Preliminaries

Let $S = \{p_1, p_2, \ldots, p_n\}$ be a set of n points in general position in the plane. We define a *convex order* of the points as follows. Let $S_i = \{p_i, p_{i+1}, \ldots, p_n\}$ for $i = 1, \ldots, n$. The order $(p_1, p_2, \ldots, p_n)$ is *convex* if every point $p_i, i = 1, \ldots, n$ lies on the boundary of the convex hull of S_i. An example of the convex order is the lexicographical order by (x, y)-coordinates. This order is used, for example,

[1]The authors conjectured that the running time is $O(\log n)$ per pseudo-triangulation.

in the incremental algorithm for constructing the convex hull. Let $x(p_i)$ and $y(p_i)$ denote the x- and y-coordinates of a point p_i.

A vertex of a polygon is *convex* if its interior angle is less than π, otherwise the vertex is *reflex*. A polygon is called a *pseudo-triangle* if it has exactly three convex vertices. A *pseudo-quadrangle* has exactly four convex vertices. In general, a *pseudo-k-gon* has exactly k convex vertices.

A *side* of a pseudo-k-gon P consists of vertices and edges between two convex vertices (the internal vertices of a side are reflex vertices of P). Let $T_1(P), T_2(P), \ldots, T_k(P)$ denote binary search trees storing the vertices of the sides of P in clockwise order. Note that every convex vertex of P participates in two trees and every reflex vertex is contained in one tree. The following lemma generalizes flips in triangulations to flips in pseudo-triangulations, see Fig. 1.

Lemma 1 (Flips in a pseudo-quadrangle) *Every pseudo-quadrangle Q has exactly two partitions into two pseudo-triangles which are produced by two diagonals. The diagonals either intersect properly or share a reflex vertex of the pseudo-quadrangle.*

The diagonals can be computed in $O(\log n)$ time if the vertices of Q are stored in four balanced search trees $T_i(Q), i = 1, 2, 3, 4$ corresponding to the sides of Q.

Proof: The existence of two diagonals has been shown by Rote *et al.* [19]. Let Q be a pseudo-quadrangle and let a, b, c, and d be the convex vertices of Q in counterclockwise order. The partitions can be obtained by cutting Q with the shortest paths from a to c and from b to d in Q (see details in Lemma 2.2 [19] and Lemma 2.1 [22]). Each shortest path has exactly one edge inside Q, which is a diagonal that splits Q into two pseudo-triangles, see Fig. 1 (b).

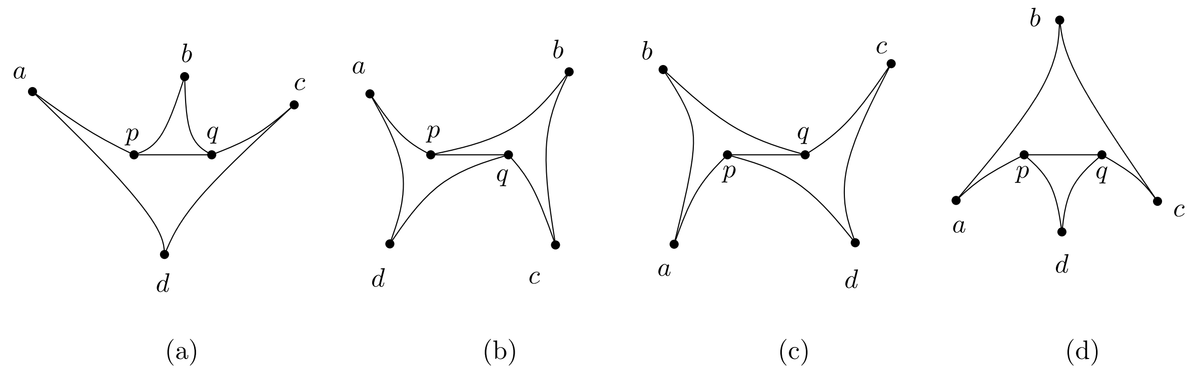

Figure 2: Four types of the shortest path between a to c.

We show that the diagonals cannot share a convex vertex. Clearly, the shortest path between a and c avoids the vertices b and d since they are convex. Similarly the shortest path between b and d avoids the vertices a and c. Thus two shortest paths ac and bd can share reflex vertices only.

We show that the diagonals intersect. Let p be q be the endpoints of the diagonal from the shortest path ac. The line pq is common tangent to two sides of Q, one is ab or ad and the other is bc or cd, see Fig. 2. Without loss of generality we can assume that the line pq is horizontal. In all four cases depicted in Fig. 2 b is above the line pq and d is below the line pq. The shortest path bd contains an edge $p'q'$ such that p' lies above or on the line pq and q' lies below or on the line pq. It can be verified that $p'q'$ is the second diagonal and the diagonals intersect.

Computing diagonals. Each diagonal is a bitangent to two sides of Q. A bitangent to two convex polygons can be computed in $O(\log n)$ time [14]. A side of Q can be completed to a convex

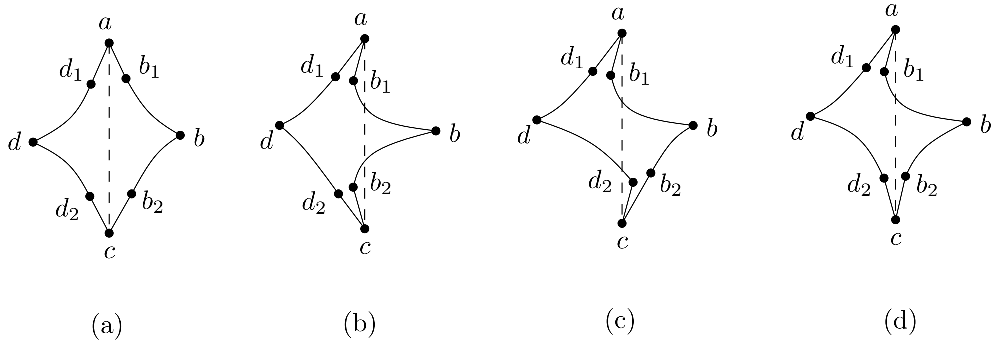

Figure 3: Finding a diagonal of Q on the shortest path ac.

polygon by adding the straight line segment between the side endpoints. Let P_{ab}, P_{bc}, P_{cd} and P_{ad} denote the polygons corresponding to the sides ab, bc, cd and ad, respectively. We show how the sides of Q can found in $O(1)$ time. Suppose that we want to find a diagonal of the geodesic ac. Suppose that the point c is below a. Let ab_1 be the edge of the side ab. Similarly we denote the edges ad_1, cb_2, cd_2, see Fig. 3.

We consider the following cases depending on the location the segment $b_id_i, i = 1, 2$ relative to the segment ac. If ac intersect both b_1d_1 and b_2d_2, then ac is the required diagonal, see Fig. 3 (a). If both b_1d_1 and b_2d_2 lie on the same side of the line ac, say on the left side (the case of the right side is symmetric), then the diagonal is a bitangent of P_{ab} and P_{bc}, see Fig. 3 (b).

Suppose that the line ac separates the segments b_1d_1 and b_2d_2, say b_1d_1 lies to the left of ac and b_2d_2 lies to the right of ac. Then the diagonal is a bitangent of P_{ab} and P_{cd}, see Fig. 3 (c).

Suppose that the line ac intersects only one of the segments b_1d_1 and b_2d_2, say b_2d_2. Suppose that b_1d_1 lies to the left of the line ac (otherwise it is symmetric). Then the diagonal is a bitangent of P_{ab} and P_{cd}, see Fig. 3 (d).

Thus the diagonals can be found in $O(\log n)$ time. ∎

The flip of an edge e in a pseudo-triangulation removes e and inserts another edge e'. We call e' a *dual edge of* e. A *pseudo-triangulation* of S is a partitioning of the convex hull of S into pseudo-triangles with vertices in S such that each point of S is a vertex of a pseudo-triangle. A pseudo-triangulation T is a *minimum pseudo-triangulation* or a *pointed pseudo-triangulation* if it contains a minimum number of edges. Streinu [20] proved that a pointed pseudo-triangulation of a n-point set contains $2n - 3$ edges.

Henneberg construction. Streinu [20] proved that the graph of a pointed pseudo-triangulation is minimally rigid. A minimally rigid graph [9, 23] can be constructed using the Henneberg updates [10]. There are two types of graph updates: (I) attach a new vertex by two edges, and (II) delete an edge and attach a new vertex to the endpoints of the deleted edge plus one other vertex. Figure 4 illustrates the Henneberg construction of type I. It can be used to generate a pointed pseudo-triangulation of S.

Let G be the *graph of pointed pseudo-triangulations* of S. Recall that its vertices correspond to pointed pseudo-triangulations and the edges correspond to flips. Similarly the graph of triangulations G_T is defined [11, 4]. These graphs share some properties though the graphs induced by the same point sets are different in general. Both graphs G_T and G are connected (the connectivity of G has been shown in [19] and follows from Theorem 3 below). They coincide if the points of S are in convex position.

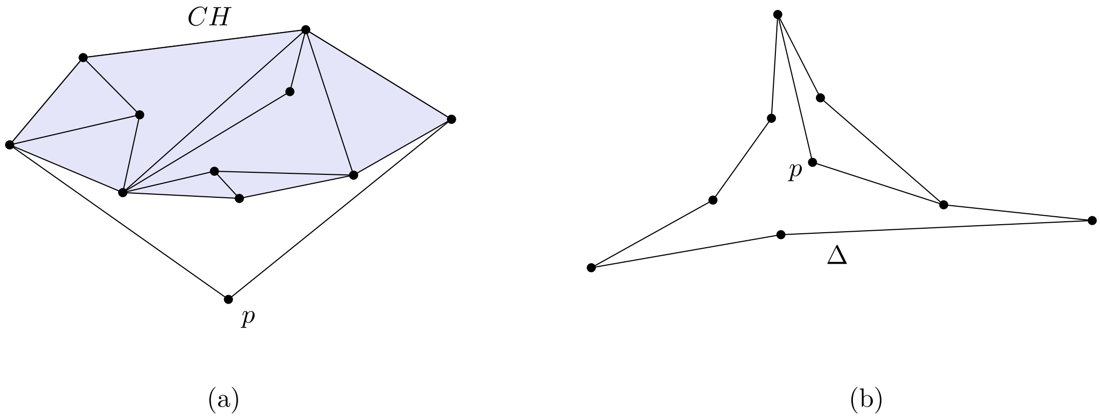

Figure 4: The Henneberg construction of type I applied for a point p lying (a) outside the convex hull, and (b) inside a pseudo-triangle.

Hurtado *et al.* [11] showed that there are $2n$ points in the plane and two triangulations that require more than $(n-1)^2$ flips to transform one into the other. Brönnimann *et al.* [6] showed that $O(n^2)$ flips suffice to transform a pointed pseudo-triangulation into another one. Interestingly, the flip distance between pseudo-triangulations is smaller than the one between triangulations: recently, we improved the bound to $O(n \log n)$ [5].

3 Tree of Pointed Pseudo-Triangulations

Let $conv(A)$ denote the convex hull of a set A. Let $(p_1, p_2, \ldots, p_n)$ be a convex order of the points in S. A pointed pseudo-triangulation T contains all the edges of $conv(S) = conv(S_1)$. We define an *index* of T denoted by $index(T)$ as the largest $k \leq n-2$ so that T has the edges of $conv(S_1), \ldots, conv(S_k)$. Clearly, $k \geq 1$ for any pointed pseudo-triangulation. The pseudo-triangulation T contains $conv(S_k)$. We define a *degree* of T denoted by $degree(T)$ as the number of pseudo-triangles of T that are contained in S_k and have p_k as a vertex. With each pointed pseudo-triangulation we associate a vector $\alpha(T) = (k, -l)$ where $k = index(T)$ and $l = degree(T)$, see Fig. 5 (a) for an example. The lexicographical order of vectors $\alpha(T)$ induces a partial order on the set of pseudo-triangulations.

Lemma 2 *There is a unique pointed pseudo-triangulation with the lexico-largest vector α.*

Proof: Let T be the pseudo-triangulation obtained by the Henneberg construction of type I applied for the points in the order $(p_n, p_{n-1}, \ldots, p_1)$: start with the triangle $p_n p_{n-1} p_{n-2}$ and insert points $p_{n-3}, \ldots, p_1$. The pseudo-triangulation T is pointed (it can be easily verified) by Theorem 3.1 [20].

We show that the vector α of T is the lexico-largest vector among all pointed pseudo-triangulations. The index of T is $n-2$ since T contains the edges of $conv(S_i), i = 1, 2, \ldots, n-2$. The index of T has the maximum value. Any pseudo-triangulation T' with index $n-2$ contains all the edges of $conv(S_i), i = 1, 2, \ldots, n-2$ and, thus, is a supergraph of T. By the definition of pointed pseudo-triangulation $T' = T$. The lemma follows. ∎

We denote by T_{max} a pseudo-triangulation with the lexico-largest vector.

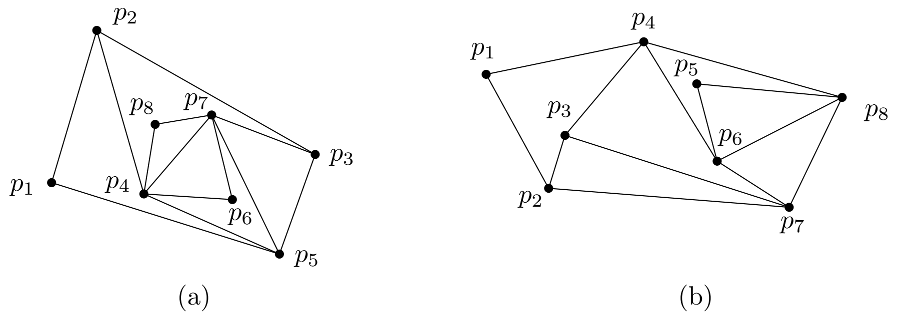

Figure 5: (a) A pseudo-triangulation T with $index(T) = 3$ and $\alpha(T) = (4, -3)$, (b) a pseudo-triangulation with the lexico-largest vector $\alpha(T) = (6, -1)$.

Theorem 3 *For any pointed pseudo-triangulation $T \neq T_{max}$, there is a flip making a pseudo-triangulation with lexico-larger vector α.*

Proof: Let $\alpha(T) = (k, -l)$ be the vector of T. Since $\alpha(T)$ is not the lexico-largest vector, $\alpha(T) \neq (n-2, -1)$ and $k < n-2$. The degree of T, l, must be greater than 1; otherwise T has all the edges of the convex hull of S_{k+1} and the first index must be less than k. In other words, there are at least two pseudo-triangles inside the convex hull of S_k that have the common vertex p_k. Let Δ_1 and Δ_2 be two such pseudo-triangles that share a common edge (p_k, p_i). Let Q be the polygon which is the union of Δ_1 and Δ_2.

We show that Q is a pseudo-quadrangle. Note that Q has at least three convex vertices p_k, p_{k_1} and p_{k_2} where p_{k_1} and p_{k_2} are the convex vertices of Δ_1 and Δ_2 different from p_k and p_i, see Fig. 6. On the other hand, there are at most 6 candidates to be convex vertices of Q (the convex vertices of Δ_1 and Δ_2). Let γ_1 and γ_2 be the angles between the segments incident to p_i in the pseudo-triangle Δ_1 and Δ_2, respectively. The angles satisfy one of the the following cases.

Case 1: $\gamma_1 + \gamma_2 < \pi$. The number of convex vertices of Q is four since p_k and p_i are the convex vertices of both Δ_1 and Δ_2, see Fig. 6 (a).

Case 2: $\gamma_1 + \gamma_2 > \pi$ and $\gamma_1, \gamma_2 < \pi$. The number of convex vertices of Q is three, see Fig. 6 (b). It contradicts the fact that T is a pointed pseudo-triangulation.

Case 3: $\gamma_1 + \gamma_2 > \pi$ and one of the angles γ_1 or γ_2 is greater than π. The number of convex vertices of Q is four, see Fig. 6 (c).

By Lemma 1 the polygon Q can be partitioned into two pseudo-triangles in two ways and corresponding diagonals do not share the vertex p_k. ∎

We show how a spanning tree $\mathcal{T}$ of G can be constructed. It is possible to define the tree using an approach similar to [4] which is based on a lexicographical order of triangulations using edge vectors (note that the pointed pseudo-triangulations have a fixed number of edges). Unfortunately, it is not clear how to traverse the resulting tree efficiently. We apply a different approach for pseudo-triangulations.

We restrict the order of points to be monotone in some direction. For simplicity, we assume that the points are sorted by x-coordinate and their x-coordinates are distinct (we can slightly rotate the set of the points to achieve this). The root of $\mathcal{T}$ corresponds to the pseudo-triangulation

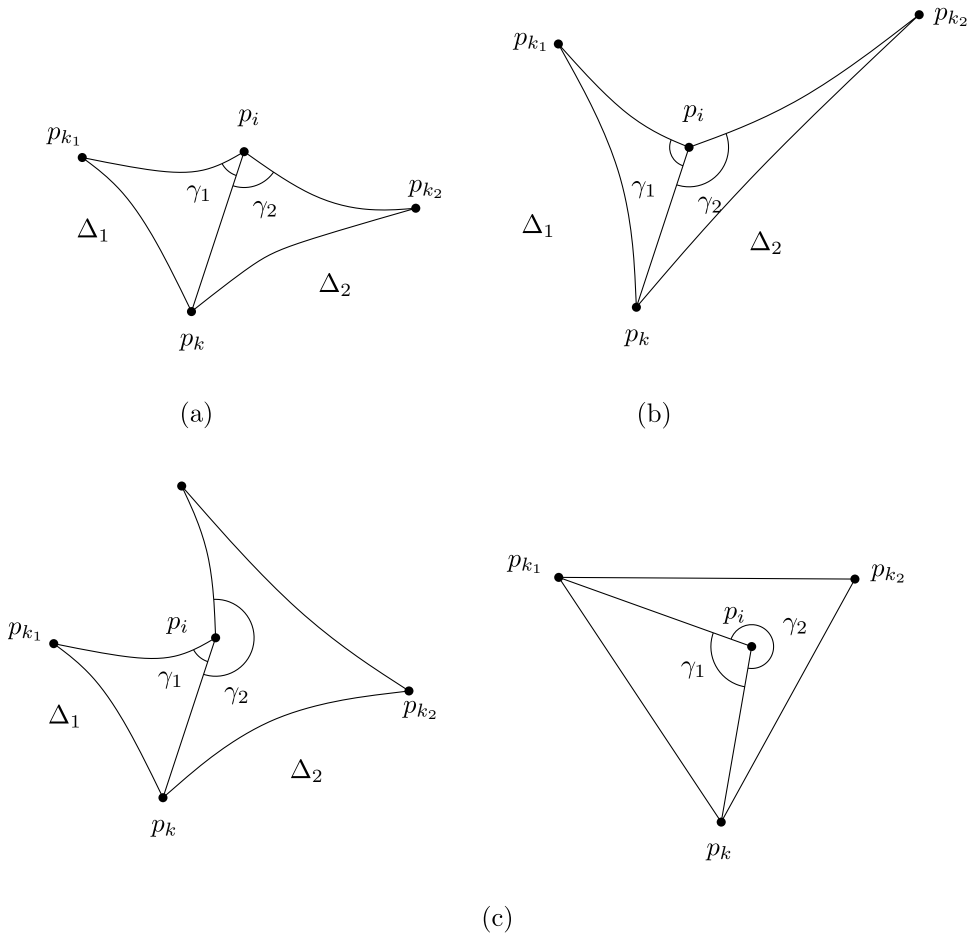

Figure 6: Cases. (a) and (c) Q is a pseudo-quadrangle. (b) Q is a pseudo-triangle.

T_{max}, see Fig. 5 (b). Let T be a pseudo-triangulation distinct from T_{max}. Theorem 3 guarantees the existence of an edge whose flip increases the vector of T. Let v be the node of G corresponding to T. We define the parent node of v as follows. Let $e = (p_a, p_b)$, $a < b$ be an edge of T such that

(i) the flip of e increases the vector $\alpha(T)$, and

(ii) a is the least number among the edges satisfying (i), and

(iii) the vector $p_a p_b$ has the maximum slope among the edges satisfying (i) and (ii).

Clearly, e is well defined. Abusing notation, we call e *parent edge* of T and edges whose flips lead to the children of v as *child edges*. The following lemma characterizes the parent edges.

Lemma 4 (Parent Edge) *An edge $e = (p_a, p_b)$, $a < b$ is the parent edge of a pseudo-triangulation T if and only if $a = index(T)$ and e has the largest slope among the edges of $T - T_{max}$.*

Proof: Let $k = index(T)$. First, we show that $a \geq k$. Suppose to the contrary that $a < k$. By the definition of $index(T)$, T includes the edges of the polygons $conv(S_a)$ and $conv(S_{a+1})$. Their difference $conv(S_a) - conv(S_{a+1})$ is a pseudo-triangle Δ, see Fig. 7. The interior of Δ does not contain edges of T since T is the pointed pseudo-triangulation. Let p_i and p_j be two adjacent vertices of $conv(S_a)$. Flip of either $p_a p_i$ or $p_a p_j$ in T destroys Δ and introduces a new edge incident to a vertex outside $conv(S_a)$. This flip decreases the vector $\alpha(T)$. The contradiction implies $a \geq k$.

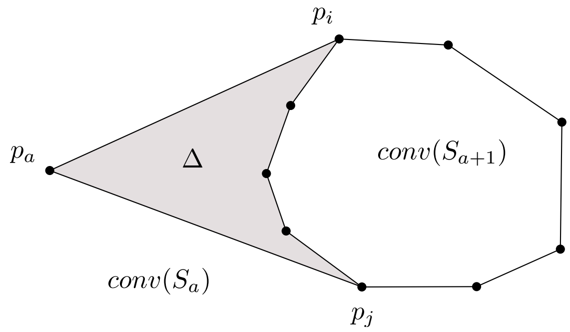

Figure 7: Pseudo-triangle Δ.

By Theorem 3 there is an edge incident to p_k satisfying the above condition (i). The condition (ii) implies $a = k$. It follows from the proof of Theorem 3 that the flip of any edge (p_a, p_b), $b > a$ of T increases $\alpha(T)$. The lemma follows. ■

The parent edge e of a pointed pseudo-triangulation T is non-vertical segment and is not an edge of the convex hull of S. Thus it is incident to two pseudo-triangles of T, one contains a nei. We denote by $\Delta(T)$ the pseudo-triangle of T that is incident to e_{parent} and is "above" e, i.e. an internal point p of the segment e enters $\Delta(T)$ by an infenitesimal motion upward. An example of $\Delta(T)$ is depicted in Fig. 9 where (p_k, p_u) is the parent edge.

Next we characterize the child edges in $\mathcal{T}$.

Theorem 5 (Child Edge) *Let v be a node of $\mathcal{T}$ and let T be its pseudo-triangulation with $\alpha(T) = (k, -l)$. Let $p_k p_j p_m$ be the convex vertices of $\Delta(T)$ in clockwise order. Let C be the side of $\Delta(T)$ between p_j and p_m. Let $p_j = p_{j_1}, p_{j_2}, \ldots, p_m$ be the vertices of the side C. An edge $e = (p_a, p_b), a < b$ is a child edge of T if and only if e is not an edge of the convex hull of S and one of the following conditions holds*

(i) *$a < k$, or*

(ii) *e is an edge of $conv(S_k)$, or*

(iii) *$\{a, b\} = \{j_r, j_{r+1}\}, m \notin \{a, b\}$ and both p_a and p_b are visible from p_k in $\Delta(T)$, or*

(iv) (a) *$\{a, b\} = \{j_r, j_{r+1}\}$ and p_{j_r} is visible from p_k in $\Delta(T)$, and*
  (b) *$j_{r+1} = m$ or $p_{j_{r+1}}$ is not visible from p_k in $\Delta(T)$, and*
  (c) *the dual edge of e is incident to p_k.*

Proof: If). (i) If $a < k$ then e is a child edge since its flip decreases the vector $\alpha(T)$, see the proof of Lemma 4.

(ii) The edge e lies on the boundary of two pseudo-triangles, say Δ_1 and Δ_2. Since e is an edge of $conv(S_k)$, one of the pseudo-triangles, say Δ_2, lies in $conv(S_k)$ and the other lies outside $conv(S_k)$, see Fig. 8. Let x be the largest number such that Δ_1 lies in $conv(S_x)$. Note that $1 \le x < k$ since Δ_1 is contained in $conv(S_1)$ but not in $conv(S_k)$. Also $\Delta_1 = conv(S_x) - conv(S_{x+1})$ since T contains all the edges of $conv(S_i), i = 1, \ldots, k$.

Let $p_x p_y p_z$ be the convex vertices of Δ_1 and let p_c be the convex vertex of Δ_2 different from p_a and p_b, see Fig. 8. The dual edge of e is contained in the shortest path between p_x and p_c in

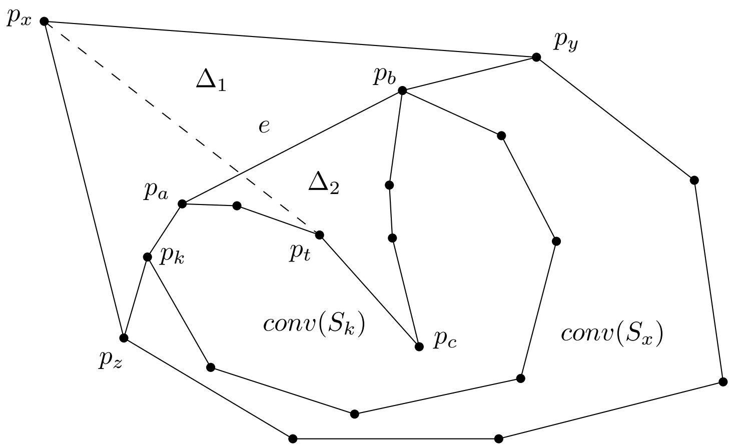

Figure 8: Pseudo-triangle Δ_1 with convex vertex p_z, $x(p_z) < x(p_k)$.

the pseudo-quadrangle $Q = \Delta_1 \cup \Delta_2$. The shortest path $p_x p_c$ avoids the convex vertices p_x and p_y of Q. Therefore the dual edge of e is incident to p_x (since p_x is adjacent to p_y and p_z in Q). Let $e' = (p_x, p_t)$ be the dual edge of e and let T' be the pseudo-triangulation obtained from T by flipping e. The edge e' is the parent edge of T' by Lemma 7.

(iii) Consider the third case. The pseudo-triangle $\Delta(T)$ contains $p_{j_r} p_{j_{r+1}}$. Let $\Delta_1 = p_{j_r} p_{j_{r+1}} p_c$ be the pseudo-triangle of T on the other side of $p_{j_r} p_{j_{r+1}}$, see Fig. 9. The union $Q = \Delta(T) \cup \Delta_1$ is a pseudo-quadrangle since p_k, p_j, p_c and p_m are the convex vertices of Q.

By Lemma 1, e and its dual edge intersect either properly or at a vertex of e. Let q be the intersection point. The edge e is completely visible from p_k. Clearly, $p_k q$ is the part of the shortest path $p_k p_c$ in the pseudo-quadrangle Q. Therefore the dual edge of e is incident to p_k. Let (p_k, p_v) be the dual edge of e. Let (p_k, p_u) be the parent edge of T, see Fig. 9.

We show that $v \notin \{u, j\}$. If $v = u$ then the vertices $p_u, p_{j_{r+1}}$ and p_m coincide. This is impossible since $j_{r+1} \neq m$. If $v = j$ then p_c is above the line passing through p_k and p_j. This is impossible since p_c is contained in $conv(S_k)$ and (p_k, p_j) is its edge.

Let T' be the pseudo-triangulation obtained by flipping e in T. The parent edge of T' is $p_k p_v$ by Lemma 7. Thus e is a child edge of T.

(iv) The forth case follows from the condition (c).

Only If). We show that e is not a child edge of T if none of the conditions (i)-(iv) holds. This assumption implies that e lies in the interior $conv(S_k)$. Let T' be the pseudo-triangulation obtained from T by flipping e, let e' be the dual edge of e and let (p_k, p_u) be the parent edge of T.

Suppose that e is not an edge of $\Delta(T)$. Then $\Delta(T)$ is a pseudo-triangle of T' and (p_k, p_u) is still the parent edge of T'. Therefore e is not a child edge.

Suppose that e is an edge of the side $p_j p_m$ of $\Delta(T)$. If both p_a and p_b are not visible from p_k, then e is not visible from p_k and p_k is not incident to e' (by Lemma 1, e' intersects e). Then e is not a child edge. If both p_a and p_b are visible from p_k and $m \notin \{a, b\}$, then it is the case (iii) and e is a child edge. In the remaining case e' is not incident to p_k by the condition (iv)(c). Therefore e is not a child edge.

Suppose that e is an edge of the side $p_k p_m$ of $\Delta(T)$. Clearly, (p_k, p_u) is not a child edge since its flip increases the vector of T. Suppose that $e \neq (p_k, p_u)$. The edge e is not on the boundary of $conv(S_k)$ by the condition (ii) (the chain $p_k p_m$ cannot actually contain an edge of $conv(S_k)$ since

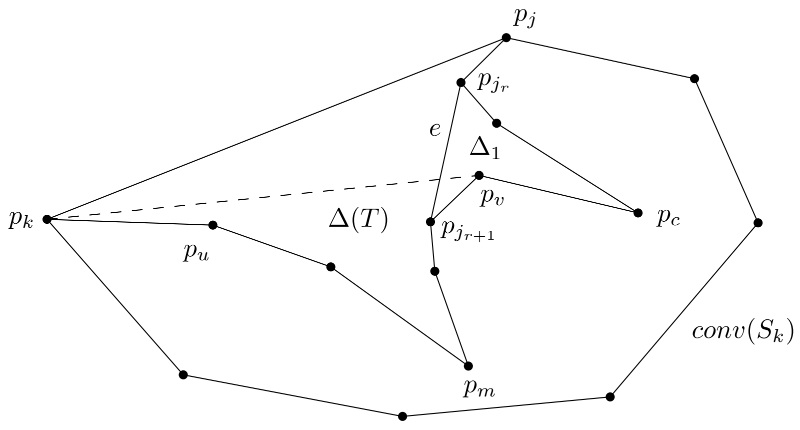

Figure 9: The dual edge of $e = (p_{j_r}, p_{j_{r+1}})$ is (p_k, p_v).

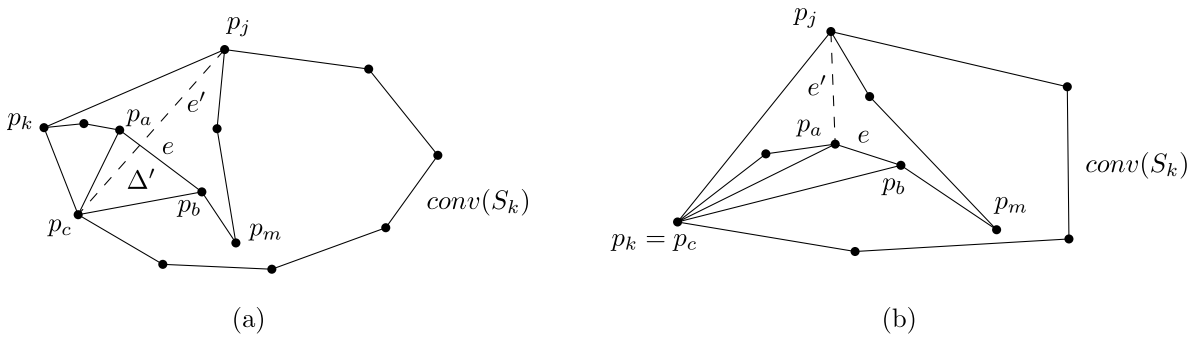

Figure 10: The edge e is not a child edge of T.

the internal vertices of p_kp_m are reflex). Let Δ' be the pseudo-triangle with e on its boundary and different from $\Delta(T)$. Let p_c be the convex vertex of Δ' opposite to the side of e, see Fig. 10. The edge e' is a part of the shortest path from p_j to p_c in the pseudo-quadrangle $Q = \Delta(T) \cup \Delta'$. We show that e' is not incident to p_k (note that p_c may coincide with p_k). If $p_c \neq p_k$, then it follows from the fact that the shortest path p_jp_c avoids the convex vertex p_k of Q, see Fig. 10 (a). If $p_c = p_k$, then it follows from $e \neq (p_k, p_u)$, see Fig. 10 (b). ■

Theorem 6 (Height of $\mathcal{T}$) *Let S be a set of n points in general position in the plane. The tree of pseudo-triangulations $\mathcal{T}$ has height at most $\binom{n-2}{2}$. The bound is tight in the worst case.*

Proof: The tree of pseudo-triangulations is consistent with the partial order on the set of pseudo-triangulations. In order to show the upper bound on the height of $\mathcal{T}$ it suffices to prove that the length of the partial order on the set of vectors $\alpha(T)$ is at most $\binom{n-2}{2}$. This follows from the fact that there are $\binom{n-2}{2} + 1$ possible vectors $\alpha() = (k, -l)$ (note that every flip changes $\alpha()$).

We count all possible vectors $\alpha()$. Let k be any integer from $1, \ldots, n-3$ and let T be a pseudo-triangulation with index k. The number of possible pseudo-triangles of T in $conv(S_k)$ incident to p_k ranges from 2 to $n-k-1$. Thus the number of vectors $\alpha()$ for the fixed k is $n-k-2$. If $k = n-2$, then the only pseudo-triangulation with index k is T_{max}. The total number of vectors

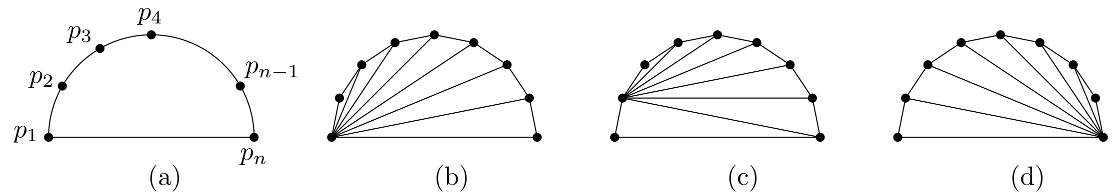

Figure 11: (a) n points on the arc. (b) Pseudo-triangulation with lexico-smallest $\alpha()$. (c) Pseudo-triangulation after $n-3$ flips. (d) T_{max}.

$\alpha()$ is

$$1 + \sum_{k=1}^{n-2} n - k - 2 = 1 + \sum_{i=1}^{n-3} i = 1 + \binom{n-2}{2}.$$

It remains to prove the lower bound. We place n points on the unit arc $x^2 + y^2 = 1, y \geq 0$, see Fig. 11 (a). The pseudo-triangulation with the lexico-smallest vector $\alpha(T) = (1, n-2)$ is depicted on Fig. 11 (b). There are $n-3$ parent flips before we obtain a pseudo-triangulation with index two, see Fig. 11 (c). In general, there are $n-k-2$ parent flips on pointed pseudo-triangulations of index k. The theorem follows. ■

An example of a spanning tree of pseudo-triangulations for a set of five points is illustrated in Fig. 12.

4 Enumerating Pointed Pseudo-Triangulations

We apply the reverse search technique by Avis and Fukuda [3]. A possible approach is to use a recursive procedure that processes a node of $\mathcal{T}$. By Theorem 6 the space requirement for this approach is $\Omega(n^2)$ in the worst case. We show that the space size can be reduced to linear.

We need some properties of child edges for an efficient algorithm. We classify the child edges according to the cases of Theorem 5, for example, the edges of type (i) correspond to the case (i). For every vertex $p_i \in S, i = 1, 2, \ldots, n-2$ we denote by $e_i^u = (p_i, p_{up(i)})$ and $e_i^l = (p_i, p_{low(i)})$ two edges of $conv(S_i)$ incident to p_i (e_i^u lies on the upper hull of S_i and e_i^l lies on the lower hull of S_i).

Lemma 7 *Let T be a pseudo-triangulation with index k.*

(i) *Let n_1 be the number of child edges of type* (i) *in T. Then $k-2 \leq n_1 \leq 2(k-2)$.*

(ii) *The child edges of type* (ii) *in T can be reported in $O(1)$ time per edge using the index functions $up()$ and $low()$.*

Proof: (i) Let $(p_a, p_b), a < b$ be a child edge of type (i). Then $2 \leq a \leq k-1$. By the definition of the index of T, there are exactly two edges (p_a, p_{b_1}) and (p_a, p_{b_2}) in T satisfying $b_i > a$. This implies the upper bound for n_1. The lower bound follows from the fact that at least one of the edges $(p_a, p_{b_i}), i = 1, 2$ lies inside the convex hull of S (since p_1 lies outside $conv(S_a)$).

(ii) The edges e_k^u and e_k^l are the edges of $conv(S_k)$ incident to p_k. Let C_{up} and C_{low} be the upper hull and the lower hull of $conv(S_k)$, respectively. We show how to find edges of C_{up}. Suppose that p_i is the current vertex of C_{up}. Note that p_i can be incident to many edges, see Fig. 13. It turns out that the next vertex in C_{up} is always $p_{up(i)}$.

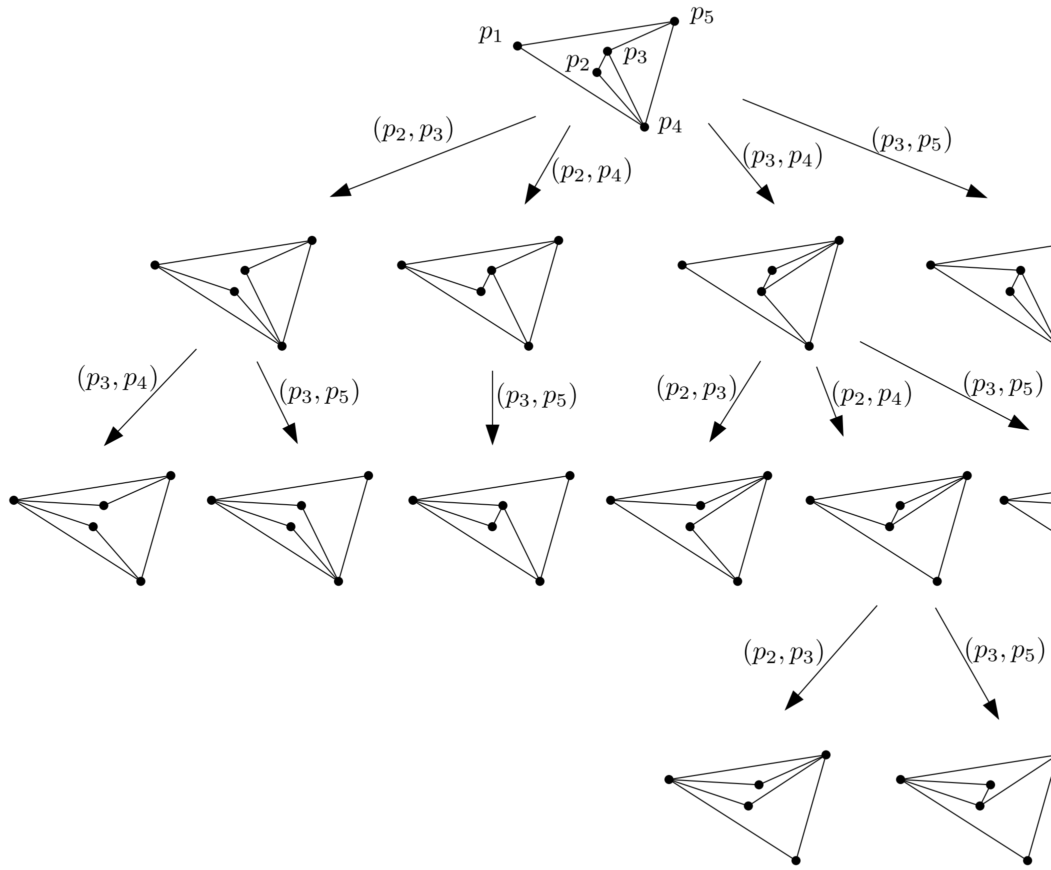

Figure 12: A spanning tree of pseudo-triangulations for five points. The labels of tree edges are the parent edges.

Let p_j be the next vertex of $conv(S_k)$, see Fig. 13. We claim that $j = up(i)$. The point $p_{up(i)}$ lies in $conv(S_k)$ since $p_{up(i)} \in conv(S_i) \subseteq conv(S_k)$. Thus, the slope of the segment p_ip_j is at least the slope of $p_ip_{up(i)}$. Note that $x(p_j) > x(p_i)$ (or $j > i$). On the other hand, the slope of $p_ip_{up(i)}$ is the maximum slope among the segments $p_ip_l, l > i$. Therefore $p_j = p_{up(i)}$. Thus, the edges of C_{up} can be found using the function $up()$. The edges of C_{low} can be found similarly. ■

Theorem 8 *Let S be a set of n points in the plane. The pointed pseudo-triangulations of S can be reported in $O(\log n)$ time per pseudo-triangulation using linear space.*

Proof: The algorithm maintains data structures that allow an efficient traversal of $\mathcal{T}$. There are static data structures that store information about the order of the points and T_{max}. The edges of the convex hull $conv(S)$ are stored in a balanced binary tree T_{conv} in the lexico-graphical order so that any edge $(p_i, p_j), i < j$ can be tested whether it is in $conv(S)$ in $O(\log n)$ time. The edges of T_{max} are stored in a binary search tree according to the lexico-graphical order. We store the values $up(i)$ and $low(i)$ using two arrays.

We store dynamic structures related to the current pseudo-triangulation T. Let L_t be the list of pseudo-triangles of T. With every side s of a pseudo-triangle Δ of T we associate a binary search tree $T_s(\Delta)$ storing its points in counterclockwise order. We also assume that the operations of

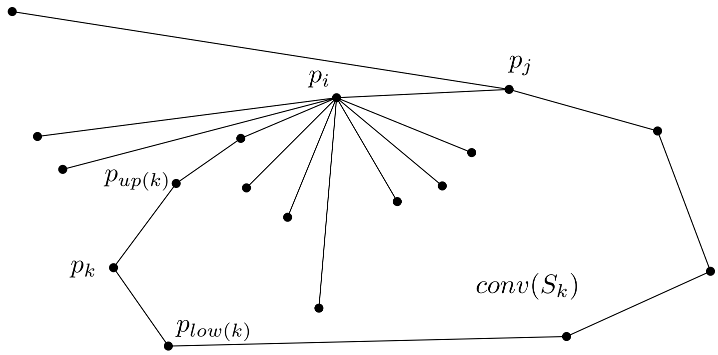

Figure 13: Finding the edges of C_{up}.

concatenation and *split* can be performed in $O(\log n)$ time using a union-split data structure [8] for example. Note that a point $p \in S$ can occur in many trees $T_s(\Delta)$ (the number of trees is actually at most the degree of p in T). The predecessor and the successor of a point $p \in T_s(\Delta)$ provide access to the edges incident to p along the side s. Every edge occurs in two sets and we store it twice with pointers to each other. We also store

(i) L_e, the list of edges in T. With every edge we store pointers to (at most) two incident pseudo-triangles.

(ii) L_-, the list of edges of $T - T_{max}$ in the lexicographical order. For every point $p \in S$ we store $L_1(p)$ the list of edges in $T - T_{max}$ incident to p in the sorted order by the slope.

We apply the reverse search technique [3] which can be viewed as a depth-first traversal of the pseudo-triangulation tree $\mathcal{T}$. For a current pseudo-triangulation T, the algorithm maintains its index, the parent edge and the pseudo-triangle $\Delta(T)$. The value of $index(T)$ can be computed by finding the smallest element in L_-. The parent edge can be found using $L_1(p_k)$. The pseudo-triangle $\Delta(T)$ can be found by checking the pseudo-triangles incident to the parent edge.

We traverse $\mathcal{T}$ as follows. The flip making a pseudo-triangulation of the parent is defined by the parent edge. The child edges can be found using Theorem 5. We describe how the child edges of each type can be found.

Case (i). We show that the child edges of type (i) can be found in $O(1)$ time per edge. The algorithm checks the vertices $p_i, i = 2, 3, \dots, k-1$ and the edges of $conv(S_i)$ incident to p_i. We use T_{conv} to test whether an edge e is a child edge. The total time is $O(k)$ and the number child edges of type (i) is $\Omega(k)$ by Lemma 7 (i).

Case (ii). The edges can be found by Lemma 7 (ii).

Cases (iii-iv). The algorithm traverses the edges of the side $p_j p_m$ of $\Delta(T)$ in the counterclockwise order (from p_j to p_m). Let $e = (p_{j_r}, p_{j_{r+1}})$ be the current edge. We can detect if p_{j_r} and $p_{j_{r+1}}$ are visible from p_k in $O(\log n)$ time. By Lemma 1 the dual edge of e can be found in $O(\log n)$ time. Thus we can detect in $O(\log n)$ time if e has type (iii) or (iv). Note that the edges of types (iii) and (iv) form a continuous path $p_j p_{j_l}$. We stop the search if either e is the last edge of $p_j p_m$ or e does not satisfy the conditions (iii) and (iv).

The linear space can be achieved using a non-recursive algorithm. For this, we maintain a boolean variable `NewNode` indicating whether the current node of T is visited for the first time. Depending on `NewNode` we find the child edge by calling `FindFirstChild()` or `FindNextChild()`, see Algorithm 1. We store the current child edge in `ChildEdge` and the next child edge in

`NewChildEdge`. Both variables store pointers to the endpoints and the type of the edge so that next child edge can be computed. Clearly, the space is linear. ■

Algorithm 1 Enumeration of pointed pseudo-triangulations.

Compute $T_{max}, conv(S), T_{conv}, up(), low()$, the lists $L_t, L_e, L_-, L_1(p)$. {Initialization}
`NewNode` = true;
loop {main loop}
  if (`NewNode`) **then** {we arrived at a new node of $\mathcal{T}$}
    `NewChildEdge = FindFirstChild();`
  else {old node of $\mathcal{T}$}
    `NewChildEdge = FindNextChild();`
  end if
  if (`NewChildEdge` <> NULL) **then** {there is a new child edge}
    `NewNode`=true;
    `ChildEdge=NewChildEdge`;
    `Output(ChildEdge);` {new pseudo-triangulation}
    `Flip(ChildEdge);`
  else {all child edges are visited}
    if (`ParentEdge` = NULL) **then** {the root of $\mathcal{T}$}
      return;
    end if
    `NewNode`=false;
    `ChildEdge=Dual(ParentEdge);`
    `Output(ParentEdge);`
    `Flip(ParentEdge);`
  end if
end loop

5 Conclusion

We presented an algorithm for enumerating the pointed pseudo-triangulations of a set of n points in the plane. The algorithm uses flips to generate pseudo-triangulations and its running time is $O(\log n)$ per pseudo-triangulation. An interesting algorithmic question is to count the number of pseudo-triangulations without generating them. This might help to verify the conjecture [6, 18] that the number of pseudo-triangulations for a finite set of points in the plane is at most the number of its triangulations.

References

[1] O. Aichholzer, F. Aurenhammer, H. Krasser, and B. Speckmann. Convexity minimizes pseudo-triangulations. In *Proc. 14th Canad. Conf. Comput. Geom.*, pp. 158–161, 2002.

[2] O. Aichholzer, G. Rote, B. Speckmann, and I. Streinu. The zigzag path of a pseudo-triangulation. In *Proc. 10th Workshop Algorithms Data Struct.*, pp. 377–389, 2003.

[3] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65:21–46, 1996.

[4] S. Bespamyatnikh. An efficient algorithm for enumeration of triangulations. *Comput. Geom. Theory Appl.*, 23(3):271–279, 2002, `http://dx.doi.org/10.1016/S0925-7721(02)00111-6`.

[5] S. Bespamyatnikh. Transforming pseudo-triangulations. In *Proc. International Conference on Computational Science*, LNCS 2657, pp. 533–539, 2003, `http://www.springerlink.com/link.asp?id=5t160n7ubnryyeku`.

[6] H. Brönnimann, L. Kettner, M. Pocchiola, and J. Snoeyink. Counting and enumerating pseudo-triangulations with the greedy flip algorithm. In *Fall Workshop on Comput. Geometry*, 2001.

[7] R. Connelly, E. D. Demaine, and G. Rote. Straightening polygonal arcs and convexifying polygonal cycles. In *Proc. 41th Annu. Sympos. on Found. of Computer Science*, pp. 432–442, 2000.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2nd edition, 2001.

[9] J. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity*, volume 2. Amer. Math. Soc., Graduate Studies in Mathematics, 1993.

[10] L. Henneberg. *Die graphische Statik der starren Systeme*. Leipzig, 1911.

[11] F. Hurtado, M. Noy, and J. Urrutia. Flipping edges in triangulations. *Discrete Comput. Geom.*, 22(3):333–346, 1999.

[12] L. Kettner, D. Kirkpatrick, and B. Speckmann. Tight degree bounds for pseudo-triangulations of points. In *Proc. 13th Canad. Conf. Comput. Geom.*, pp. 117–120, 2001.

[13] D. Kirkpatrick, J. Snoeyink, and B. Speckmann. Kinetic collision detection for simple polygons. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pp. 322–330, 2000.

[14] M. H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *J. Comput. Syst. Sci.*, 23:166–204, 1981.

[15] M. Pocchiola and G. Vegter. Pseudo-triangulations: Theory and applications. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pp. 291–300, 1996.

[16] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudo-triangulations. *Discrete Comput. Geom.*, 16(4):419–453, 1996.

[17] M. Pocchiola and G. Vegter. The visibility complex. *Int. J. Comput. Geom. Appl.*, 6(3):279–308, 1996.

[18] D. Randall, G. Rote, F. Santos, and J. Snoeyink. Counting triangulations and pseudo-triangulations of wheels. In *Proc. 13th Canad. Conf. Comput. Geom.*, pp. 149–152, 2001.

[19] G. Rote, F. Santos, and I. Streinu. Expansive motions and the polytope of pointed pseudo-triangulations. *Discr. Comput. Geom. - The Goodman-Pollack Festschrift*, pp. 699–736, 2003, `http://arxiv.org/abs/math.CO/0206027`.

[20] I. Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *Proc. 41st Annu. IEEE Sympos. Found. Comput. Sci.*, pp. 443–453, 2000, `http://cs.smith.edu/~streinu/Papers/motion.ps.gz`.

[21] I. Streinu. Folding carpenter's rulers, robot arms, proteins: a rigidity theoretic approach. *Invited talk, 10th Annual Fall Workshop on Computational Geometry*, 2000.

[22] I. Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. preprint, 2002.

[23] W. Whiteley. *Matroids from Discrete Geometry.* in Matroid Theory, J. Bonin, J. Oxley and B. Servatius (eds.), AMS Contemporary Mathematics, 1997.