Computational Geometry: Theory and Applications 128 (2025) 102171

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

journal homepage: www.elsevier.com/locate/comgeo

Finding a largest-area triangle in a terrain in near-linear time ☆

Sergio Cabello [a,b,*], Arun Kumar Das [c], Sandip Das [c], Joydeep Mukherjee [d]

[a] Faculty of Mathematics and Physics, University of Ljubljana, Slovenia
[b] Institute of Mathematics, Physics and Mechanics, Slovenia
[c] Advanced Computing and Microelectronics Unit, Indian Statistical Institute, India
[d] Dept. of Computer Science, Ramakrishna Mission Vivekananda Educational and Research Institute, India

ARTICLE INFO

Article history:
Received 13 February 2024
Received in revised form 26 November 2024
Accepted 13 February 2025
Available online 17 February 2025

Keywords:
Terrain
Inclusion problem
Geometric optimization
Hereditary segment tree

ABSTRACT

A terrain is an x-monotone polygon whose lower boundary is a single line segment. We present an algorithm to find in a terrain a triangle of largest area in $O(n \log n)$ time, where n is the number of vertices defining the terrain. The best previous algorithm for this problem has a running time of $O(n^2)$.
© 2025 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

An *inclusion problem* asks to find a geometric object inside a given polygon that is optimal with respect to a certain parameter of interest. This parameter can be the area, the perimeter or any other measure of the inner object that plays a role in the application at hand. Several variants of the inclusion problem come up depending on the parameter to optimize, the constraints imposed in the sought object, as well as the assumptions we can make about the containing polygon. For example, computing a largest-area or largest-perimeter convex polygon inside a given polygon is quite well studied [5,15,17]. A significant amount of work has also been done on computing largest-area triangle inside a given polygon [3,7,13,20]. In the last few years, there have been new efficient algorithms for the problems of finding a largest-area triangle [23,18], a largest-area or a largest-perimeter rectangle [4], and a largest-area quadrilateral [21] inside a given convex polygon. In this paper, we propose a deterministic $O(n \log n)$-time algorithm to find a largest-area triangle inside a given terrain, which improves the best known running time of $O(n^2)$, presented in [10]. These problems find applications in stock cutting [8], robot motion planning [22], occlusion culling [17] and many other domains of facility location and operational research. In general, geometric optimization problems, such as *inclusion, enclosure, packing, covering, and location-allocation*, often are considered in the area of Operations Research [2,12,19].

A polygon P is *x-monotone* if it has no vertical edge and each vertical line intersects P in an interval, which may be empty. An x-monotone polygon has a unique vertex with locally minimum x-coordinate, that is, a vertex whose two adjacent vertices have larger x-coordinate; see for example [11, Lemma 3.4]. Similarly, it has a unique vertex with locally maximum x-coordinate. If we split the boundary of an x-monotone polygon at the unique vertices with maximum and minimum x-

☆ A preliminary version of this work was presented at WADS 2021 [6].

* Corresponding author.
E-mail addresses: sergio.cabello@fmf.uni-lj.si (S. Cabello), arund426@gmail.com (A.K. Das), sandipdas@isical.ac.in (S. Das), joydeep.m1981@gmail.com (J. Mukherjee).

https://doi.org/10.1016/j.comgeo.2025.102171
0925-7721/© 2025 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

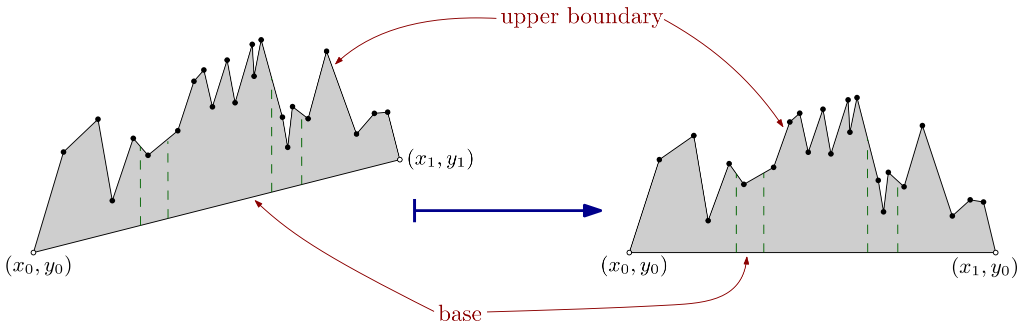

Fig. 1. Two terrains. The right one is obtained from the left one by a shear transformation to make the base horizontal.

coordinate, we get the *upper boundary* and the *lower boundary* of the polygon. Each vertical line intersects each of those boundaries at most once.

A *terrain* is an x-monotone polygon whose lower boundary is a single line segment, called the *base* of the terrain. The upper boundary of the terrain connects the endpoints of the base and lies above the base: each vertical ray from the base upwards intersects the upper boundary at exactly one point. Fig. 1 shows two examples of terrains.

In this work, we focus on the problem of finding a triangle of largest area inside a terrain. We will show that when the terrain has n vertices, such a largest-area triangle can be computed in $O(n \log n)$ time. This is an improvement over the algorithm of Das et al. [10], which has a running time of $O(n^2)$. It should be noted that we compute a single triangle with largest area, even if there are more optimal solutions.

To obtain our new algorithm we build on the approach and geometric insights of [10]. More precisely, in that work, there is a single type of optimal solution that takes $O(n^2)$ time, while all the other cases can be handled in $O(n \log n)$ time. We show that the remaining case also can be solved in $O(n \log n)$ time combining shortest path trees in polygons [16], hereditary segment trees [9], search for row maxima in monotone matrices [1], and additional geometric insights.

Our new time bound, $O(n \log n)$, is a significant improvement over the best previous result. Nevertheless, it could be that the problem is solvable in linear time. We only know that the problem cannot be solved in sublinear time because we need to scan all the vertices of the polygon. Indeed, any vertex of the terrain that is not scanned could be arbitrarily high and be the top vertex of a triangle with arbitrarily large area. Even if we assume that, say, the base and the highest vertex is also specified with the input, or the vertex that is furthest vertically from the base, it could be that the second highest vertex is the top vertex of the triangle of largest area. We leave closing this gap between $O(n \log n)$ and $\Omega(n)$ as an interesting open problem for future research.

2. Preliminaries

Without loss of generality, we will assume that *the base of the terrain is horizontal*. The general case reduces to this one, as follows. If the endpoints of the base are (x_0, y_0) and (x_1, y_1) with $x_0 \neq x_1$, then the shear mapping $(x, y) \mapsto \left(x,\ y - (x - x_0)\frac{y_1 - y_0}{x_1 - x_0}\right)$ transforms the base to the horizontal segment connecting (x_0, y_0) to (x_1, y_0). Since the mapping also transforms each vertical segment to a vertical segment, the terrain gets mapped to a terrain with a horizontal base. See Fig. 1. Because the determinant of the Jacobian matrix of this affine transformation is 1, the area of any measurable region of the plane is invariant under the transformation. Therefore, it suffices to find the triangle of largest area in the resulting polygon.

For simplicity in the description of the algorithm, we will assume the following general position: *no three vertices in the terrain are collinear*. This property is also invariant under shear transformations. The assumption can be lifted using simulation of simplicity [14]. More precisely, we can assume that each vertex $v_i = (x_i, y_i)$ is replaced by a vertex $v_i' = (x_i, y_i + \epsilon^i)$ for a sufficiently small $\epsilon > 0$. These transformations break all collinearities if ϵ is sufficiently small. The replacement is not actually performed, but simulated. More precisely, whenever the vertices v_i, v_j and v_k are collinear, which means that

$$\begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix} = 0,$$

the relative position of v_i', v_j' and v_k', after the replacements, is given by the sign of the determinant

$$\begin{vmatrix} 1 & x_i & y_i + \epsilon^i \\ 1 & x_j & y_j + \epsilon^j \\ 1 & x_k & y_k + \epsilon^k \end{vmatrix} = \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix} + \begin{vmatrix} 1 & x_i & \epsilon^i \\ 1 & x_j & \epsilon^j \\ 1 & x_k & \epsilon^k \end{vmatrix} = \begin{vmatrix} 1 & x_i & \epsilon^i \\ 1 & x_j & \epsilon^j \\ 1 & x_k & \epsilon^k \end{vmatrix}$$

$$= \epsilon^i \cdot \begin{vmatrix} 1 & x_j \\ 1 & x_k \end{vmatrix} - \epsilon^j \cdot \begin{vmatrix} 1 & x_i \\ 1 & x_k \end{vmatrix} + \epsilon^k \cdot \begin{vmatrix} 1 & x_i \\ 1 & x_j \end{vmatrix}$$

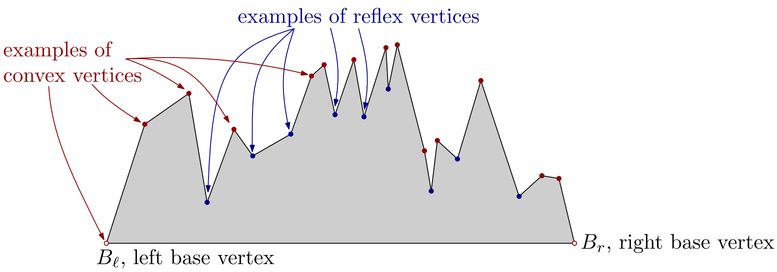

Fig. 2. Notation for vertices of the terrain.

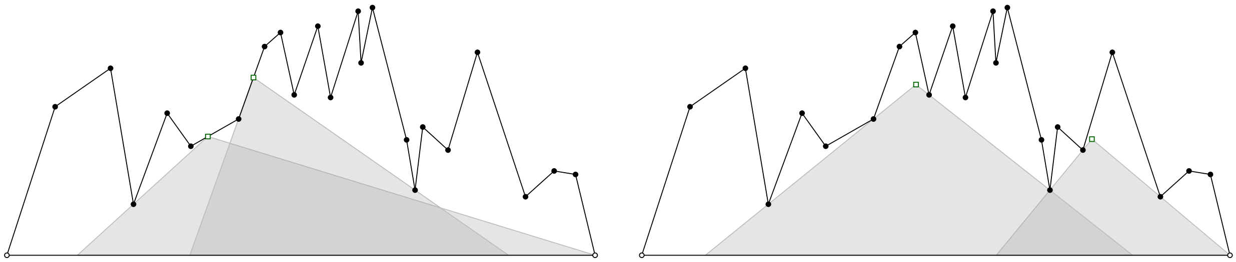

Fig. 3. Examples of grounded triangles like those in Lemma 1. The apex is marked with a square and may lie on the boundary of the terrain (left) or not (right).

$$= \epsilon^i(x_k - x_j) - \epsilon^j(x_k - x_i) + \epsilon^k(x_j - x_i).$$

For example, if $i < j$ and $i < k$, then $\epsilon^i \gg \epsilon^j$ and $\epsilon^i \gg \epsilon^k$ whenever ϵ is positive and sufficiently small. Since the x-coordinates of vertices of a terrain are distinct, we get in this case, $i < \min\{j,k\}$, that the relative position of v'_i, v'_j and v'_k is given by the sign of $x_k - x_j$. The other cases are similar.

Finally, let us introduce some notation for the vertices. See Fig. 2. A vertex of a terrain is *convex* if the internal angle between the edges incident to this vertex is less than π radians. If the angle is greater than π radians, then the vertex is *reflex*. Angles of π radians do not occur because of our assumption of no 3 collinear points. The endpoints of the base of the terrain are called *base vertices*. The one with smallest x-coordinate is the *left base vertex* and is denoted by B_ℓ. The base vertex with largest x-coordinate is the *right base vertex* and is denoted by B_r. The base vertices are convex.

3. Previous geometric observations

In this section, we state several observations and properties given in [10], without repeating their proofs here. The first one shows that to search for an optimal solution we can restrict our attention to certain types of triangles.

A triangle contained in the terrain with an edge on the base of the terrain is a *grounded triangle*. For a grounded triangle, the vertex not contained in the base of the terrain is the *apex*, and the edges incident to the apex are the *left side* and the *right side*; the right side is incident to the vertex of the base with larger x-coordinate.

Lemma 1 *(Lemmas 1 and 2, Corollary 1 in [10]). In each terrain there is a largest area triangle T satisfying all of the following properties:*

(a) *the triangle T is grounded;*
(b) *the apex of T lies on the boundary of the terrain* or *each of the left and right sides of T contains two vertices of the terrain.*

From now on, we restrict our attention to triangles that satisfy the properties of Lemma 1, and select one with largest area. Note that property (b) splits into two cases; see Fig. 3. An option is that the apex of the grounded triangle is on the boundary of the terrain. The other option is that each of the edges incident to the apex contains two vertices of the terrain; those vertices may be base vertices. The first case is already solved in $O(n \log n)$ time; this is the content of the following lemma.

Lemma 2 *(Implicit in [10]; see the paragraph before Theorem 1). Given a terrain with n vertices, we can find in $O(n \log n)$ time the grounded triangle with largest area that has its apex on the boundary of the terrain.*

The key insight to obtain Lemma 2 is to decompose the upper boundary of the terrain into $O(n)$ pieces with the following property: for any two points p, q in the same piece of the upper boundary, the largest grounded triangle with

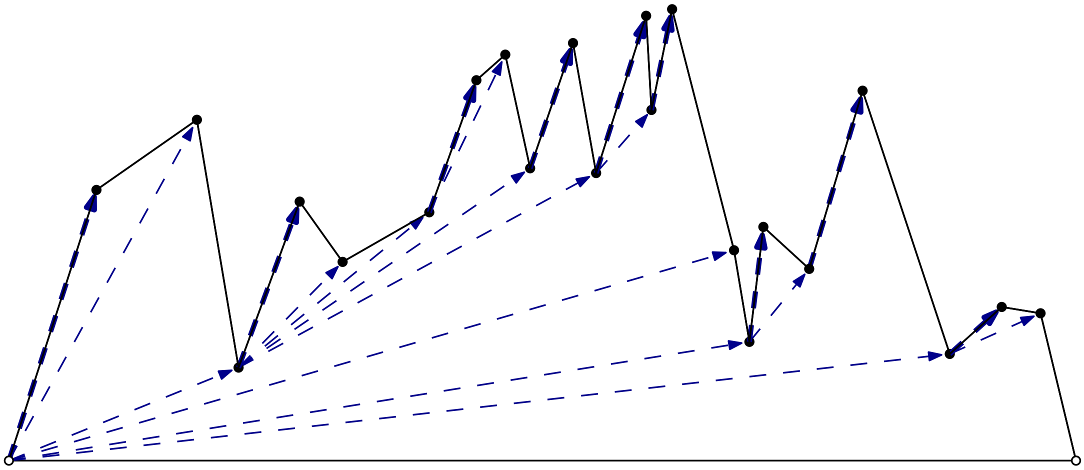

Fig. 4. The tree T_ℓ with blue dashed arcs. Edges of T_ℓ contained in an edge of the polygon are thicker to make them visible. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

apex at p and the largest grounded triangle with apex at q have the same vertices of the terrain on their left sides and the same vertices of the terrain on their right sides. When the piece is not a single point, there is precisely one vertex on each side of the triangle, and together with the apex they define uniquely a grounded triangle. One can then write a closed formula for the area of the largest grounded triangle when the apex moves along a piece, and find its maximum in $O(1)$ time per piece. We refer to [10] for further details.

It remains to solve the case where the apex is *not* contained on the boundary of the terrain. This means that each side of the optimal triangle contains two vertices of the terrain. Because the triangle is grounded, there are two options for each of the sides: either both vertices contained in a side are reflex vertices, or one vertex is reflex and the other is a vertex of the base.

Recall that B_ℓ is the left endpoint of the base of the terrain. Consider the visibility graph of the vertices of the terrain and let T_ℓ be the shortest path tree from the vertex B_ℓ in the visibility graph. The set of edges of T_ℓ are denoted by $E(T_\ell)$ and we orient them away from the root, consistent with the direction that the shortest path from B_ℓ would follow them. We regard T_ℓ indistinctly as a graph and as a geometric object, that is, a set of oriented segments connecting vertices of the terrain. See Fig. 4 for an example.

Consider an (oriented) edge $p \to q$ of T_ℓ; the point p is closer to B_ℓ than q is. It may be that $p = B_\ell$. The *forward prolongation* of $p \to q$ is the segment obtained by extending the directed segment $p \to q$ inside the terrain until it reaches the boundary of the terrain. The interior of the segment pq is not part of the prolongation. The forward prolongation may be empty. However, if the forward prolongation of $p \to q$ is non-empty, then q is a reflex vertex and the line supporting $p \to q$ is tangent to the boundary of the polygon at q. Each point on the forward prolongation is further from B_ℓ than q is. Let L be the set of non-zero-length forward prolongations of segments $p \to q$ of T_ℓ. See Fig. 5. The *backward prolongation* of an edge $p \to q$ of T_ℓ is the extension of $p \to q$ from p in the direction $q \to p$ until it reaches the boundary of the terrain. The backward prolongation of $p \to q$ is empty if and only if $p = B_\ell$.

A similar construction is done to obtain a shortest-path tree T_r from the right endpoint B_r of the base of the terrain, the prolongations of its edges, and the set R of non-empty forward prolongations for the edges of T_r. See Fig. 6. Similarly, we use $E(T_r)$ to denote the set of edges of T_r, which may be treated as a set of oriented segments.

Using that the terrain is an x-monotone polygon and the lower boundary is a single segment, one obtains the following properties.

Lemma 3 *(Lemmas 3, 4 and 5 in [10]). The backward prolongation of each edge in $E(T_\ell) \cup E(T_r)$ has an endpoint on the base of the terrain; it may be a vertex of the base. The segments in $E(T_\ell)$ have positive slope and the segments in $E(T_r)$ have negative slope.*

If the apex of the grounded triangle with largest area is not *on the boundary of the terrain, then there is an edge s_ℓ of L and an edge s_r of R such that: the left side of the triangle is collinear with s_ℓ, the right side of the triangle is collinear with s_r, and the apex of the triangle is the intersection $s_\ell \cap s_r$.*

4. New algorithm

We are now going to describe the new algorithm. In fact, we describe the missing piece in the previous approach of [10]. Because of Lemma 1, it suffices to search for the grounded triangle of largest area. We have two cases to consider: the apex may be on the boundary of the terrain or not. The first case can be handled using Lemma 2. To approach the second case, we use Lemma 3: in such a case the apex of the triangle belongs to $A = \{s_\ell \cap s_r \mid s_\ell \in L,\ s_r \in R\}$. We refer to A as the *set of*

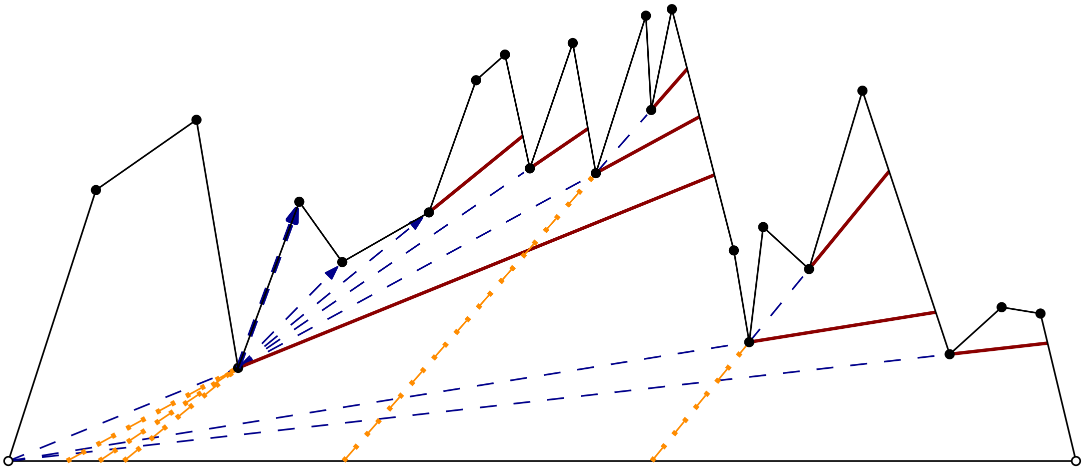

Fig. 5. Forward and some backward prolongations in the example of Fig. 4. The set L, of non-empty forward prolongations of the edges of T_ℓ, is in solid, thick red. The edges of T_ℓ defining L are in dashed blue. In dashed-dotted orange is the set of backward prolongations for the edges defining L.

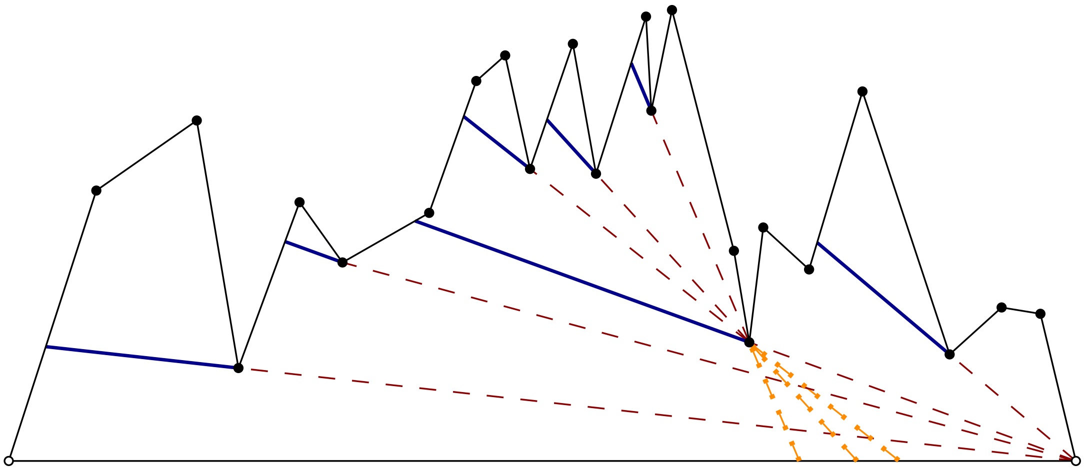

Fig. 6. Forward and some backward prolongations for T_r in the example of Fig. 4. The set R, of non-empty forward prolongations of the edges of T_r, is in solid, thick blue. The edges of T_r defining R are in dashed red. In dashed-dotted orange is the set of backward prolongations for the edges defining R.

candidate apices. See Fig. 7. Each candidate apex $s_\ell \cap s_r$ defines uniquely a triangle because its sides must be collinear with s_ℓ, s_r and the base of the terrain.

We start providing a simple property for L and R.

Lemma 4. *The edges of L are pairwise interior-disjoint and can be computed in $O(n)$ time. The same holds for R.*

Proof. Consider the forward prolongation $qt \in L$ of the oriented edge $p \to q$ of T_ℓ. The shortest path from B_ℓ to any point on qt consists of the shortest path from B_ℓ to q followed by a portion of qt. (See for example [16, Lemma 2.1] for the structure of shortest paths in a polygon without holes.) It follows that the edges of L are contained in shortest paths from B_ℓ and thus they are pairwise disjoint. (They cannot overlap because of our assumption on general position.)

Guibas et al. [16] show how to compute in $O(n)$ time the shortest path tree T_ℓ from B_ℓ and the forward extensions L. This is the *extended algorithm* discussed after their Theorem 2.1, where they decompose the polygon into regions such that the shortest path to any point in the region goes through the same sequence of vertices of the polygon. $\square$

We use Lemma 4 to compute L and R in linear time. We will start using ℓ or ℓ_i for a generic segment of L and r or r_j for a generic segment of R. Note that $L \cup R$ has $O(n)$ segments. However, the set A of candidate apices may have quadratic size, as can be seen in the schematic construction of Fig. 8. To get our improved running time we treat them implicitly.

We use a *hereditary segment tree*, introduced by Chazelle et al. [9], as follows. We decompose the x-axis into intervals using the x-coordinates of the endpoints of the segments in $L \cup R$. We disregard the two unbounded intervals, that is, the leftmost and the rightmost intervals. The resulting intervals are called the *atomic intervals*. See Fig. 9 for an example where

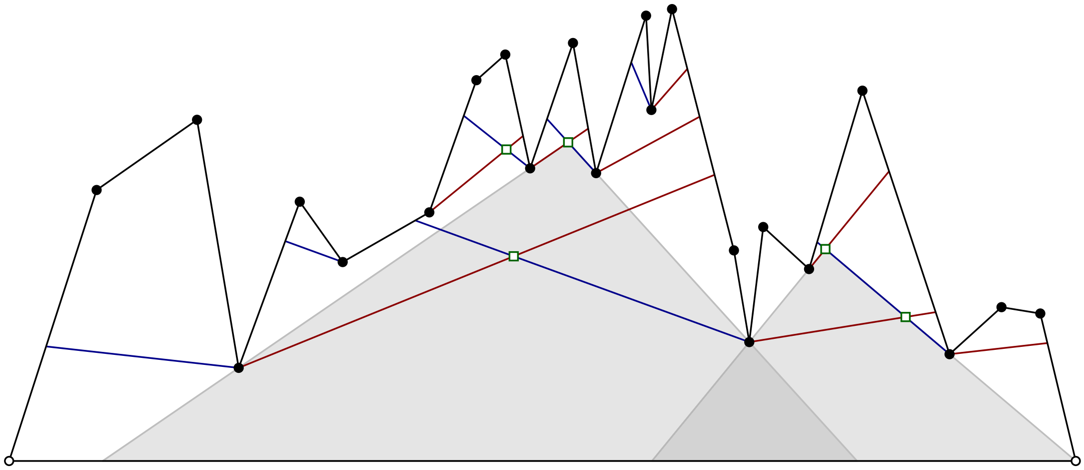

Fig. 7. The candidate apices for the example of Fig. 4 are marked with green boxes. Two of the triangles they define are shaded.

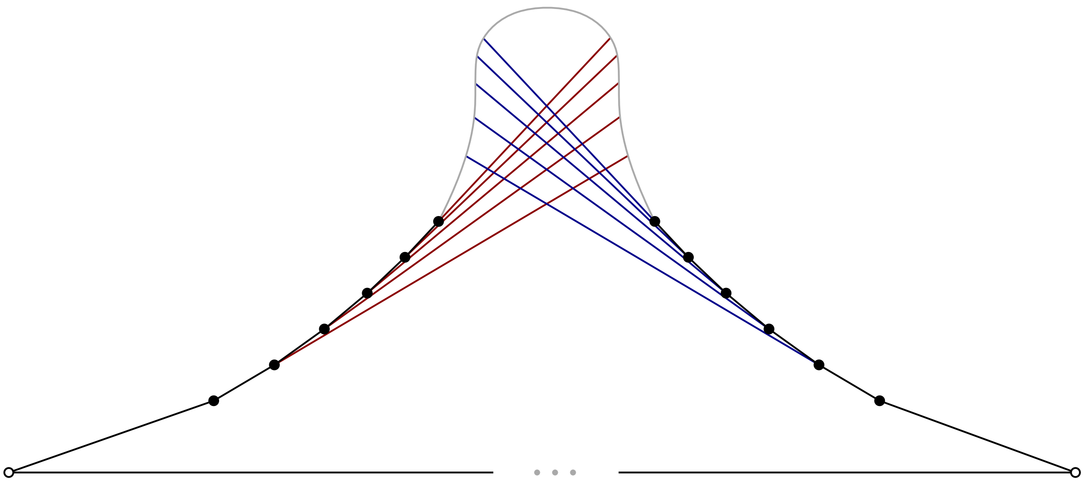

Fig. 8. Example showing that we may have a quadratic number of candidate apices.

the atomic intervals are marked as $1, 2, \ldots, 11$. We make a height-balanced binary tree $\mathcal{T}$ such that the i-th leaf represents the i-th atomic interval from left to right; see Fig. 10. For each node v of $\mathcal{T}$, we define the interval I_v as the union of all the intervals represented in the leaves of the subtree rooted at v. Alternatively, for each internal node v, the interval I_v is the union of the intervals represented by its two children. In the two-dimensional setting, the node v represents the vertical strip bounded by the vertical lines passing through the end points of I_v. Let us denote this strip by J_v. In Fig. 9, the vertical strip J_v is highlighted for the node v of Fig. 10 that is highlighted.

Consider a node v of $\mathcal{T}$ and denote by w its parent. We maintain in v four lists of segments: L_v, R_v, L_v^h and R_v^h. The list L_v contains all the segments $\ell \in L$ such that the x-projection of ℓ contains I_v but does not contain I_w. Similarly, R_v contains the segments $r \in R$ whose projection onto the x-axis contains I_v but does not contain I_w. We call L_v and R_v the *standard* lists. The list L_v^h contains the members of L_u for all proper descendants u of v in $\mathcal{T}$, that is, all descendants of v excluding v itself. Similarly, R_v^h contains the members of R_u for all proper descendants u of v in $\mathcal{T}$. We call L_v^h and R_v^h the *hereditary* lists. We put only one copy of a segment in a hereditary list of a node, even if it is stored in more than one of its descendants. See Fig. 10 for an example. The standard lists and the hereditary lists for a node are stored explicitly at the node.

Chazelle et al. [9] noted that

$$\sum_{v \text{ node of } \mathcal{T}} \left(|L_v| + |R_v| + |L_v^h| + |R_v^h|\right) = O(n \log n). \tag{1}$$

To obtain this bound, one first argues that each single segment s of $L \cup R$ appears in $O(\log n)$ standard lists, namely in at most two nodes at each level. Moreover, the nodes that contain the segment s in their standard lists have $O(\log n)$ ancestors

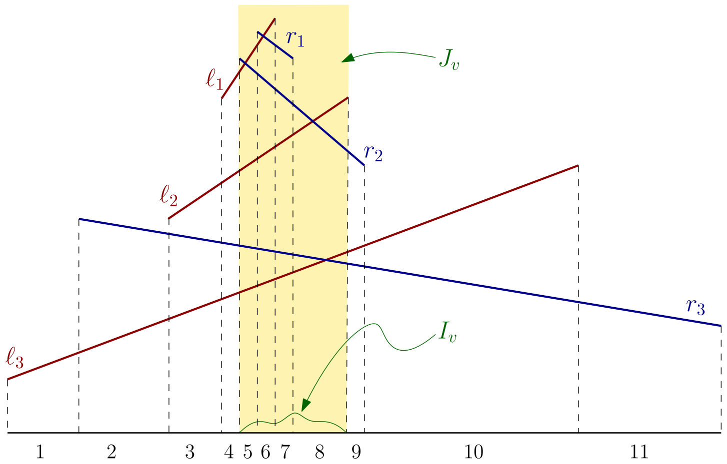

Fig. 9. Example showing the atomic intervals for a case with $|R| = |L| = 3$.

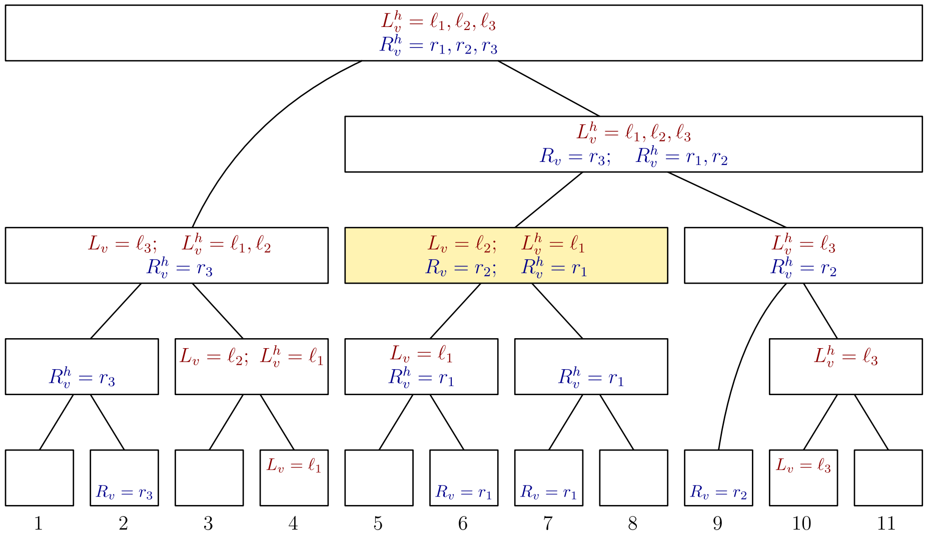

Fig. 10. Hereditary segment tree for the example of Fig. 9. All the lists that are not indicated are empty.

in total, namely a subset of the nodes on the search path in $\mathcal{T}$ to the extreme atomic intervals contained in the projection of s. It follows that each segment $s \in L \cup R$ appears in $O(\log n)$ hereditary lists.

For each node v of $\mathcal{T}$ we define the intersections

$$\begin{aligned} A_v &= \{\ell \cap r \mid \ell \in L_v,\, r \in R_v,\, x(\ell \cap r) \in I_v\} \cup \\ &\quad \{\ell \cap r \mid \ell \in L_v^h,\, r \in R_v,\, x(\ell \cap r) \in I_v\} \cup \\ &\quad \{\ell \cap r \mid \ell \in L_v,\, r \in R_v^h,\, x(\ell \cap r) \in I_v\}. \end{aligned}$$

The set A_v is the set of *candidate apices defined by the node* v. Note that in the definition of A_v we exclude the case when $\ell \in L_v^h$ and $r \in R_v^h$.

Lemma 5. *The set of candidate apices, A, is the disjoint union of the sets A_v, where v iterates over the nodes of $\mathcal{T}$.*

Proof. Consider a candidate apex $a \in A$. Because the segments of L (and R) are interior disjoint by Lemma 4, there is exactly one segment $\ell \in L$ and one segment $r \in R$ that intersect and give the apex $a = \ell \cap r$. Let u be the leaf of $\mathcal{T}$ such that the x-coordinate of $a = \ell \cap r$ is contained in I_u. Let π be the path in $\mathcal{T}$ from u to the root of $\mathcal{T}$.

We walk the path π from u upwards until the first node v with the property that $\ell \in L_v \cup L_v^h$ and $r \in R_v \cup R_v^h$ is reached. Since at the root r we have $L_r^h = L$ and $R_r^h = R$, such a node v exists. We next argue that it cannot be that $\ell \in L_v^h$ and $r \in R_v^h$. To see this, we first note that, if v is a leaf, their hereditary lists are empty. Therefore, if $(\ell, r) \in L_v^h \times R_v^h$, then v is not a leaf.

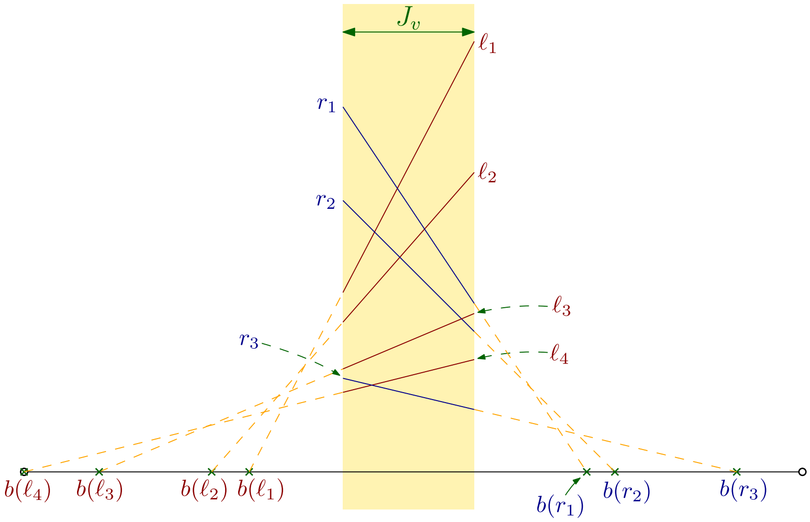

Fig. 11. Interaction between two standard lists. The segments in solid may be longer, but the part contained in J_v is contained in the segment. The dashed portion may contain part of the segment and contains the whole backward prolongation of the segment.

Consider the child w of v in the path π. Since the interval I_w of such a child w contains $x(\ell \cap r)$, the interval I_w intersects $x(\ell)$. Moreover, because $\ell \in L_v^h$, we conclude that $\ell \in L_w \cup L_w^h$. Similarly, $r \in R_w \cup R_w^h$, and thus v was not the lowest node on π with the desired property. We conclude that, indeed, $(\ell, r) \notin L_v^h \times R_v^h$. Finally, we note that the intersection point $a = \ell \cap r$ has its x-coordinate in I_v because $x(\ell \cap r) \in I_u \subseteq I_v$. It follows that $a = \ell \cap r \in A_v$.

To argue that the union is disjoint, we first note that only nodes w along π satisfy that $x(\ell \cap r) \in I_w$. Therefore, only for the nodes w of π can A_w contain $\ell \cap r$. For each proper descendant w of v along π, we have $\ell \notin L_w$ or $r \notin R_w$ because of the definition of v as the lowest node of π satisfying $\ell \in L_v \cup L_v^h$ and $r \in R_v \cup R_v^h$. Therefore, $a = \ell \cap r$ does not belong to A_w for any descendant w of v along π. For each ancestor w of v along π, we will have $\ell \in L_w^h$ and $r \in R_w^h$, and therefore because of the definition of A_w we have $a \notin A_w$ for those ancestors. $\square$

We have to find the best apex in A. Since $A = \bigcup_v A_v$ because of Lemma 5, it suffices to find the best apex in A_v for each v. (We do not use the property that the union is disjoint.) For this we consider each v separately and look at the interaction between the lists L_v and R_v, the lists L_v^h and R_v, and the lists R_v^h and L_v.

4.1. Interaction between two standard lists

Consider a fixed node v and its standard lists L_v and R_v. The x-projection of each segment in $L_v \cup R_v$ is a superset of the interval I_v, and thus no endpoint of such a segment lies in the interior of J_v. See Fig. 11 to follow the discussion in this section.

Since the segments in L_v are pairwise interior-disjoint (Lemma 4) and they cross the vertical strip J_v from left to right, we can sort them with respect to their y-order within the vertical strip J_v. We sort them in decreasing y-order. Henceforth, we regard L_v as a sorted list. Thus, L_v contains $\ell_1, \dots, \ell_{|L_v|}$ and, whenever $1 \le i < j \le |L_v|$, the segment ℓ_i is above ℓ_j. We do the same for R_v, also by decreasing y-coordinate. Thus, R_v is a sorted list $r_1, \dots, r_{|R_v|}$ and, whenever $1 \le i < j \le |R_v|$, the segment r_i is above r_j.

We will discuss in Section 4.3 how the sorted lists L_v and R_v for all nodes v together can be obtained in $O(|L| \log |L| + |R| \log |R|) = O(n \log n)$ time. For the time being, we assume that L_v and R_v are already sorted.

Because of Lemma 3, each segment s of $L \cup R$ can be prolonged inside the terrain until it hits the base of the terrain. Indeed, such a prolongation contains an edge of $E(T_\ell) \cup E(T_r)$ by definition. Let $b(s)$ be the point where the prolongation of s intersects the base of the terrain.

Lemma 6. *If $1 \le i < j \le |L_v|$, then $b(\ell_i)$ lies to the right of $b(\ell_j)$. If $1 \le i < j \le |R_v|$, then $b(r_i)$ lies to the left of $b(r_j)$.*

Proof. Let s_i be the longest segment that contains ℓ_i and is contained in the terrain; let s_j be the longest segment that contains ℓ_j and is contained in the terrain. Thus $b(\ell_i)$ is an endpoint of s_i and $b(\ell_j)$ is an endpoint of s_j. Assume, for the sake of reaching a contradiction, that $b(\ell_i)$ lies to the left of $b(\ell_j)$. This means s_i and s_j do not intersect to the left of J_v, and thus s_i is completely above s_j for any x-coordinate to the left of I_v. Then s_j cannot go through any vertex of the terrain to the left of J_v, as such a vertex would be below s_i, which is contained in the terrain. By construction of the hereditary segment tree, I_v is a proper subset of the x-projections of ℓ_j and none of the endpoints of ℓ_j belongs to the interior of J_v.

This means that the left endpoint of the segment ℓ_j should be to the left of J_v. Since the left endpoint of ℓ_j has to be a vertex of the terrain, we arrive at the contradiction.

The argument for segments of R is similar. $\square$

Once L_v and R_v are sorted, we can detect in $O(|L_v|+|R_v|)$ time which segments of L_v do not cross any segment of R_v inside J_v. Indeed, we can merge the lists to obtain the order π_ℓ of $L_v \cup R_v$ along the left boundary of J_v and the order π_r along the right boundary of J_v. Then we note that ℓ_i does not intersect any segment of R_v inside J_v if and only if the rank of ℓ_i is the same in π_ℓ and in π_r. We remove from L_v the segments that do not cross any segment of R_v inside J_v. To avoid introducing additional notation, we keep denoting to the resulting list as L_v.

Within the same running time $O(|L_v|+|R_v|)$ we can find for each $\ell_i \in L_v$ an index $\psi(i)$ such that ℓ_i and $r_{\psi(i)}$ intersect inside J_v. Indeed, if ℓ_i crosses some segment of R_v inside J_v, then because ℓ_i has positive slope and the segments of R_v have negative slope, ℓ_i must cross the segment of R_v that in the order π_ℓ is above ℓ_i. These segments for all ℓ_i can be computed with a scan of the order π_ℓ. In the example of Fig. 11, we would have $\psi(1)=\psi(2)=\psi(3)=2$ and $\psi(4)=3$.

Consider the $|L_v|\times|R_v|$ matrix $M=(M[i,j])_{i,j}$ defined as follows. If ℓ_i and r_j intersect in J_v, then $M[i,j]$ is the area of the grounded triangle with apex $\ell_i \cap r_j$ and sides containing ℓ_i and r_j. If ℓ_i and r_j do not intersect in J_v, and $j<\psi(i)$, then $M[i,j]=j\varepsilon$ for an infinitesimal $\varepsilon>0$. In the remaining case, when ℓ_i and r_j do not intersect in J_v but $\psi(i)<j$, then $M[i,j]=-j\varepsilon$ for the same infinitesimal $\varepsilon>0$. The value $\varepsilon>0$ can be treated symbolically and does not need to take any explicit value; it is only important that $n\varepsilon$ is positive and smaller than any area of any triangle we consider. (Recall that ℓ_i and $r_{\psi(i)}$ intersect, so there is no need to consider the option where ℓ_i and $r_{\psi(i)}$ do not intersect.)

Note that within each row of M the non-infinitesimal elements are contiguous. Indeed, whether a segment of L_v and a segment of R_v intersect in J_v depends only on the orders π_ℓ and π_r along the boundaries of J_v, which is the same as the order along the lists L_v and R_v. It also follows that the entries of M defined as areas of triangles form a staircase such that in lower rows it moves towards the right. It follows that a row of M, when we walk it from left to right, has small positive increasing values until it reaches values defined by the area of triangles, and then it starts taking small negative values that decrease.

The matrix M is not constructed explicitly, but we work with it implicitly. Given a pair of indices (i,j), we can compute $M[i,j]$ in constant time. In the example of Fig. 11, if we denote by $\alpha(\ell_i,r_j)$ the area of the grounded triangle with apex $\ell_i \cap r_j$ and sides containing ℓ_i and r_j, when the intersection $\ell_i \cap r_j$ exists, then we have

$$M = \begin{bmatrix} \alpha(\ell_1,r_1) & \alpha(\ell_1,r_2) & -3\varepsilon \\ \alpha(\ell_2,r_1) & \alpha(\ell_2,r_2) & -3\varepsilon \\ \varepsilon & \alpha(\ell_3,r_2) & -3\varepsilon \\ \varepsilon & 2\varepsilon & \alpha(\ell_4,r_3) \end{bmatrix}.$$

We will show that the matrix M is *totally monotone*. We start with the following special case, which is the one involving more geometry.

Lemma 7. *Consider indices i,i',j,j' such that $1\le i<i'\le|L_v|$, $1\le j<j'\le|R_v|$ and such that the four intersections $\ell_i\cap r_j$, $\ell_{i'}\cap r_j$, $\ell_i\cap r_{j'}$, $\ell_{i'}\cap r_{j'}$ occur inside J_v. If $M[i',j]>M[i',j']$, then $M[i,j]>M[i,j']$.*

Proof. See Fig. 12 to follow the proof. Because of Lemma 6, the extensions of ℓ_i and $\ell_{i'}$ inside the terrain intersect in a point to the left of J_v, which we denote by p_ℓ. Similarly, the extensions of r_j and $r_{j'}$ intersect in a point p_r to the right of J_v.

For each $(\alpha,\beta)\in\{i,i'\}\times\{j,j'\}$, let $a_{\alpha,\beta}$ be the intersection point of ℓ_α and r_β. Thus, we have defined four points, namely $a_{i,j}, a_{i,j'}, a_{i',j}, a_{i',j'}$, and by assumption they are inside J_v.

We define the following areas

$$\begin{aligned} A_1 &= \mathrm{area}\big(\triangle(b(\ell_{i'}), b(\ell_i), p_\ell,)\big), & A_2 &= \mathrm{area}\big(\triangle(p_\ell, a_{i',j'}, a_{i,j'})\big)\\ A_3 &= \mathrm{area}\big(\diamond(a_{i,j'}, a_{i',j'}, a_{i',j}, a_{i,j})\big), & A_4 &= \mathrm{area}\big(\pentagon(b(\ell_i), b(r_j), p_r, a_{i',j'}, p_\ell)\big)\\ A_5 &= \mathrm{area}\big(\triangle(a_{i',j'}, p_r, a_{i',j})\big), & A_6 &= \mathrm{area}\big(\triangle(b(r_j), b(r_{j'}), p_r)\big). \end{aligned}$$

The condition $M[i',j]>M[i',j']$ translates into

$$A_1+A_4+A_5 \;=\; M[i',j] \;>\; M[i',j'] \;=\; A_1+A_4+A_6,$$

which implies that $A_5>A_6$. We then have

$$M[i,j] \;=\; A_2+A_3+A_4+A_5 \;>\; A_2+A_3+A_4+A_6 \;>\; A_2+A_4+A_6 \;=\; M[i,j'],$$

as we wanted to show. $\square$

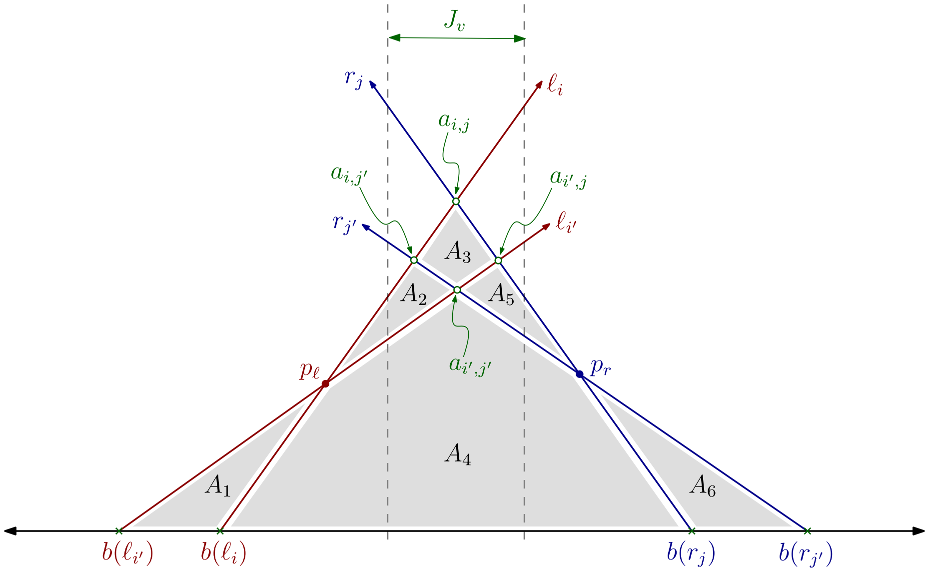

Fig. 12. Scenario in the proof of Lemma 7.

We now consider the general case to show that the matrix M is *totally monotone*. In fact, the lemma restates the definition of totally monotone matrix.

Lemma 8. *Consider indices i, i', j, j' such that $1 \le i < i' \le |L_v|$ and $1 \le j < j' \le |R_v|$. If $M[i', j] > M[i', j']$, then $M[i, j] > M[i, j']$.*

Proof. If $\ell_{i'}$ and r_j do not intersect inside J_v, then $M[i', j] > M[i', j']$ can only occur when $\ell_{i'}$ and $r_{j'}$ do not intersect inside J_v and $\psi(i') < j'$. In this case, within the strip J_v we have ℓ_i above $\ell_{i'}$ and above $r_{j'}$. This means that ℓ_i cannot intersect $r_{j'}$ inside J_v and we must have also $\psi(i) < j'$. We conclude that $M[i, j] > M[i, j']$.

If ℓ_i and $r_{j'}$ do not intersect inside J_v, we use the contrapositive. Assuming that $M[i, j] \le M[i, j']$, then because of the order within the ith row it must be that ℓ_i does not intersect r_j nor $r_{j'}$ inside J_v and $j < j' < \psi(i)$. That is, in this case, within J_v the segment r_j is above $r_{j'}$, which is above ℓ_i. Since $\ell_{i'}$ is within J_v below ℓ_i, we then have $M[i', j'] = M[i, j'] \ge M[i, j] = M[i', j]$.

It remains to consider the case where $\ell_{i'}$ and r_j intersect inside J_v and also ℓ_i and $r_{j'}$ intersect inside J_v. Using that ℓ_i is above $\ell_{i'}$, that r_j is above $r_{j'}$, that $\ell_{i'}$ intersects r_j inside J_v, and that ℓ_i intersects $r_{j'}$ inside J_v, we conclude that ℓ_i also intersects r_j inside J_v and that $\ell_{i'}$ also intersects $r_{j'}$ inside J_v. For this we just have to observe the relative order of the endpoints of the segments restricted to the boundaries of J_v. Equivalently, we get that the entries $M[i, j]$ and $M[i', j']$ are defined by areas of triangles because the staircase is moving to the right when we look at lower rows. In this case, the conclusion follows from Lemma 7. $\square$

For each index i with $1 \le i \le |L_v|$, let $\varphi(i)$ be the smallest index of columns where the maximum in the ith row of M is attained. Thus, $M[i, \varphi(i)] = \max\{M[i, j] \mid 1 \le j \le |R_v|\}$. Since M is totally monotone, we can compute the values $\varphi(i)$ for all $1 \le i \le |L_v|$ simultaneously using the SMAWK algorithm of Aggarwal et al. [1]. This step takes $O(|L_v| + |R_v|)$ time.

For the node v of $\mathcal{T}$, we return the maximum among the values $M[i, \varphi(i)]$ where $i = 1, \dots, |L_v|$. In total we have spent $O(|L_v| + |R_v|)$ time, assuming that L_v and R_v were already in sorted form.

4.2. Interaction between a standard list and a hereditary list

Consider now a fixed node v, its standard list L_v and its hereditary list R_v^h. The x-projection of each segment in L_v is a superset of the interval I_v, and thus no endpoint of such a segment lies in the interior of J_v. However, the x-projection of a segment in R_v^h has nonempty intersection with the interval I_v, but it is not a superset of I_v. This implies that each segment of R_v^h has at least one of its endpoints in the interior of J_v. See Fig. 13 to follow the discussion in this section.

Like before, we assume that the members of L_v are sorted by decreasing y-order within J_v and use $\ell_1, \dots, \ell_{|L_v|}$ to denote them from top to bottom. See Fig. 13. We will see in Section 4.3 how such an order is obtained in $O(|L_v|)$ amortized time per node v.

Lemma 9. *Each segment r from the hereditary list R_v^h has the following properties.*

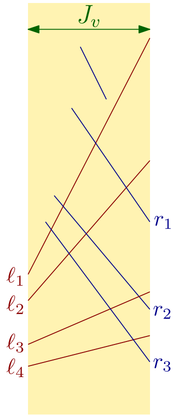

Fig. 13. Interaction between the standard list L_v and the hereditary list R_v^h. The segments in solid may be longer, but the part contained in J_v is as shown.

(a) *If r intersects $\ell \in L_v$ inside J_v, then r intersects the right boundary of J_v and, at the right boundary of J_v, r is below ℓ.*
(b) *If r intersects ℓ_i inside J_v, then for each $i' \leq i$ the segment r intersects $\ell_{i'}$ inside J_v.*

Proof. First we show that, if an endpoint of $r \in R$ is in the strip J_v, then it must be above all segments $\ell \in L_v$. For this, we note that both endpoints of r are on the boundary of the terrain: the rightmost endpoint is a vertex of the terrain and the leftmost endpoint is on the boundary of the terrain by definition. Since the terrain is x-monotone, it follows that the vertical upwards rays from both endpoints of r are outside the terrain, while the segment ℓ is contained in the terrain.

Now we show the property in (a). Assume that r intersects $\ell \in L_v$ inside J_v. For the sake of reaching a contradiction, assume that r has its right endpoint inside J_v. Then, because the slope of r is negative, the slope of ℓ is positive, and the x-projection of ℓ covers I_v, the right endpoint of r would be below ℓ and inside J_v. This contradicts the property in the previous paragraph. We conclude that r intersects the right boundary of J_v and, because of the slopes of r and ℓ, at the right boundary of J_v the segment ℓ is above r.

For property (b), it suffices to show the claim for $i' = 1$: if r intersects ℓ_i and ℓ_1 inside J_v, then it intersects $\ell_{i'}$ for all i' with $i' \leq i$. Because of (a), r intersects the right boundary of J_v and, at the right boundary of J_v, ℓ_i is above r. It follows that, at the right boundary of J_v, the segment ℓ_1 is also above ℓ_1. Because r is in the hereditary list R_v^h, at least one of the endpoints of r is inside J_v, and because of property (a), this must be the left endpoint of r. Because of the property in the first paragraph, the left endpoint of r must be above ℓ_1. It follows that the segments r and ℓ_1 cross inside J_v. □

We proceed as follows. Because of Lemma 9(a), we can discard any segment of R_v^h that does not intersect the right boundary of J_v or that is above ℓ_1 at the right boundary of J_v; they do not intersect any segment of L_v. If no elements of R_v^h remain, then no segment of L_v intersects any segment of R_v^h inside J_v and we have finished. Otherwise, let $r_1, \ldots, r_k$ be the remaining segments of R_v^h, sorted from top to bottom at the right boundary of J_v. We assume for the time being that the order is available; we will discuss later, in Section 4.3, how such an order is obtained in $O(k)$ amortized time. We can also prune the segments of L_v that do not intersect r_k; because of Lemma 9(b), if they do not intersect r_k inside J_v, they do not intersect any remaining segment of R_v^h inside J_v. Let us keep using L_v for the resulting set, to avoid introducing additional notation. Recall Fig. 13.

We now use an argument similar to the one for the standard lists in Section 4.1. We consider a $|L_v| \times k$ matrix $M = (M[i,j])_{i,j}$ defined as follows. If ℓ_i and r_j intersect in J_v, then $M[i,j]$ is the area of the grounded triangle with apex $\ell_i \cap r_j$ and sides containing ℓ_i and r_j. If ℓ_i and r_j do not intersect in J_v, then $M[i,j] = j\varepsilon$ for an infinitesimal $\varepsilon > 0$. This finishes the description of M; there is no need to consider different cases because r_k crosses all remaining segments of L_v. At each row of M, we always have some areas of triangles at the right side because r_k intersects inside J_v all remaining segments of L_v. In the example of Fig. 13, if we denote by $\alpha(\ell_i, r_j)$ the area of the triangle with apex $\ell_i \cap r_j$, when the intersection exists, then we have

$$M = \begin{bmatrix} \alpha(\ell_1,r_1) & \alpha(\ell_1,r_2) & \alpha(\ell_1,r_3) \\ \alpha(\ell_2,r_1) & \alpha(\ell_2,r_2) & \alpha(\ell_2,r_3) \\ \varepsilon & \alpha(\ell_3,r_2) & \alpha(\ell_3,r_3) \\ \varepsilon & 2\varepsilon & \alpha(\ell_4,r_3) \end{bmatrix}.$$

The same argument that was used to prove Lemma 8 shows that this matrix M is also totally monotone.

Lemma 10. *Consider indices i, i', j, j' such that $1 \leq i < i' \leq |L_v|$ and $1 \leq j < j' \leq k$. If $M[i',j] > M[i',j']$, then $M[i,j] > M[i,j']$.*

Proof. The same arguments as in the proof of Lemma 8 work. One only needs to notice that, whenever $M[i', j]$ is defined by an area, because $\ell_{i'}$ and r_j intersect, then Lemma 9 implies that, for all $j' > j$, the segment $\ell_{i'}$ also intersects $r_{j'}$ inside J_v and the segment ℓ_i intersects both r_j and r'_j inside J_v. That is, when $M[i', j]$ is defined by an area, then $M[i, j]$ is defined by an area for each $i \le i'$ and each $j' \ge j$. This implies that the crossings used in the proof of Lemma 8 exist also in this setting, and thus we can again use Lemma 7. □

Using again the SMAWK algorithm of Aggarwal et al. [1], we can find a largest area grounded triangle with apices given by the members of the sorted lists L_v and R_v^h in $O(|L_v| + |R_v^h|)$ time. We can also handle the interaction between R_v and L_v^h in a similar fashion. This finishes the description of the interaction between a standard and a hereditary list in a node v.

4.3. Putting things together

At each node v of $\mathcal{T}$, we handle the interactions between the standard lists L_v, R_v, as discussed in Section 4.1, and twice the interactions between a standard list and a hereditary list (L_v, R_v^h is one group, and L_v^h, R_v is another group), as discussed in Section 4.2. Taking the maximum over all nodes of $\mathcal{T}$, we find an optimal triangle whose apex lies in A because of Lemma 5. At each node v we spend $O(|L_v| + |R_v| + |L_v^h| + |R_v^h|)$, assuming that the lists are already sorted.

To get the lists sorted at each node, we use the same technique that Chazelle et al. [9, Section 3.1] used to improve their running time. We define the following binary relation $\prec$ on the segments in L: for any two segments $\ell, \ell' \in L$, we have $\ell \prec \ell'$ if and only if there is a vertical line that intersects the interior of ℓ and ℓ' and, along that vertical line, ℓ is immediately below ℓ' (there are no other segments of L in between). Using a left-to-right sweep-line algorithm we can compute in $O(|L| \log |L|)$ time the relations in $\prec$. Moreover, we can extend in $O(|L|)$ time this relation $\prec$ to a total order on L using a topological sort. Once this total order for L is computed at the root of the hereditary segment tree, it can be passed to its descendants in time proportional to the lengths of the lists. The restriction of this order to any node v of $\mathcal{T}$ gives the desired order for L_v, L_v^h. A similar computation is done for R separately.

Theorem 11. *A triangle of largest area inside a terrain with n vertices can be found in $O(n \log n)$ time.*

Proof. The computation of the total order extending the above-below relation takes $O(n \log n)$ for L and for R. After this, we can pass the sorted lists to each child in time proportional to the size of the lists. Thus, we spend additional $O(|L_v| + |R_v| + |L_v^h| + R_v^h|)$ time per node v of the hereditary tree to get the sorted lists.

Once the lists at each node v of the hereditary tree are sorted, we spend $O(|L_v| + |R_v| + |L_v^h| + R_v^h|)$ time to handle the apices of A_v, as explained above. Using the bound in equation (1), the total time over all nodes together is $O(n \log n)$. □

5. Conclusions

We have provided an algorithm to find a triangle of largest area contained in a terrain described by n vertices in $O(n \log n)$ time. The obvious open problem is whether the problem can be solved in linear time. It would also be interesting to identify other classes of polygons where the largest-area triangle can be computed in near-linear time.

Funding

Funded in part by The Slovenian Research and Innovation Agency (P1-0297, J1-9109, J1-1693, J1-2452, N1-0218). Funded in part by the European Union (ERC, KARST, 101071836). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] A. Aggarwal, M. Klawe, S. Moran, P. Shor, R. Wilber, Geometric applications of a matrix-searching algorithm, Algorithmica 2 (1987) 195–208.
[2] G. Altay, M.H. Akyüz, T. Öncan, Solving a minisum single facility location problem in three regions with different norms, Ann. Oper. Res. 321 (1) (2023) 1–37.
[3] J.E. Boyce, D.P. Dobkin, R.L.S. Drysdale III, L.J. Guibas, Finding extremal polygons, SIAM J. Comput. 14 (1) (1985) 134–147.

[4] S. Cabello, O. Cheong, C. Knauer, L. Schlipf, Finding largest rectangles in convex polygons, Comput. Geom. Theory Appl. 51 (2016) 67–74.
[5] S. Cabello, J. Cibulka, J. Kynčl, M. Saumell, P. Valtr, Peeling potatoes near-optimally in near-linear time, SIAM J. Comput. 46 (5) (2017) 1574–1602.
[6] S. Cabello, A.K. Das, S. Das, J. Mukherjee, Finding a largest-area triangle in a terrain in near-linear time, in: Proceedings of the 17th International Symposium Algorithms and Data Structures, WADS 2021, in: Lecture Notes in Computer Science, vol. 12808, Springer, 2021, pp. 258–270.
[7] S. Chandran, D. Mount, A parallel algorithm for enclosed and enclosing triangles, Int. J. Comput. Geom. Appl. 2 (1992) 191–214.
[8] C.K. Chang, J.S. Yap, A polynomial solution for the potato-peeling problem, Discrete Comput. Geom. 1 (2) (1986) 155–182.
[9] B. Chazelle, H. Edelsbrunner, L. Guibas, M. Sharir, Algorithms for bichromatic line-segment problems and polyhedral terrains, Algorithmica 11 (1994) 116–132.
[10] A.K. Das, S. Das, J. Mukherjee, Largest triangle inside a terrain, Theor. Comput. Sci. 858 (2021) 90–99.
[11] M. de Berg, O. Cheong, M.J. van Kreveld, M.H. Overmars, Computational Geometry: Algorithms and Applications, 3rd edition, Springer, 2008.
[12] P.M. Dearing, M.E. Cawood, The minimum covering Euclidean ball of a set of Euclidean balls in $\mathbb{R}^n$, Ann. Oper. Res. 322 (2) (2023) 631–659.
[13] D.P. Dobkin, L. Snyder, On a general method for maximizing and minimizing among certain geometric problems, in: Proceedings of the 20th Annual Symposium on Foundations of Computer Science, FOCS 1979, 1979, pp. 9–17.
[14] H. Edelsbrunner, E.P. Mücke, Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms, ACM Trans. Graph. 9 (1) (1990) 66–104.
[15] J.E. Goodman, On the largest convex polygon contained in a non-convex n-gon, or how to peel a potato, Geom. Dedic. 11 (1) (1981) 99–106.
[16] L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, R.E. Tarjan, Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons, Algorithmica 2 (1987) 209–233.
[17] O. Hall-Holt, M.J. Katz, P. Kumar, J.S.B. Mitchell, A. Sityon, Finding large sticks and potatoes in polygons, in: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, 2006, pp. 474–483.
[18] Y. Kallus, A linear-time algorithm for the maximum-area inscribed triangle in a convex polygon, Preprint available at https://arxiv.org/abs/1706.03049, 2017.
[19] R.E. Korf, M.D. Moffitt, M.E. Pollack, Optimal rectangle packing, Ann. Oper. Res. 179 (1) (2010) 261–295.
[20] E. Melissaratos, D. Souvaine, Shortest paths help solve geometric optimization problems in planar regions, SIAM J. Comput. 21 (4) (1992) 601–638.
[21] G. Rote, The largest contained quadrilateral and the smallest enclosing parallelogram of a convex polygon, Preprint at https://arxiv.org/abs/1905.11203, 2019.
[22] C.D. Tóth, J. O'Rourke, J.E. Goodman, Handbook of Discrete and Computational Geometry, CRC Press, 2017.
[23] I. van der Hoog, V. Keikha, M. Löffler, A. Mohades, J. Urhausen, Maximum-area triangle in a convex polygon, revisited, Inf. Process. Lett. 161 (2020) 105943.