Spanning Trees Crossing Few Barriers

Tetsuo Asano[1] Mark de Berg[2] Otfried Cheong[2]
Leonidas J. Guibas[3] Jack Snoeyink[4] Hisao Tamaki[5]

Abstract

We consider the problem of finding low-cost spanning trees for sets of n points in the plane, where the cost of a spanning tree is defined as the total number of intersections of tree edges with a given set of m barriers. We obtain the following results:

(i) if the barriers are possibly intersecting line segments, then there is always a spanning tree of cost $O(\min(m^2, m\sqrt{n}))$;

(ii) if the barriers are disjoint line segments, then there is always a spanning tree of cost $O(m)$;

(iii) if the barriers are disjoint convex objects, then there is always a spanning tree of cost $O(n + m)$.

All our bounds are worst-case optimal.

1 Introduction

Consider a problem of *batched point location*: efficiently locating n given points in a planar subdivision defined by m line segments. This problem arises in many applications, but particularly in work that gives a linear-time reconstruction of previously-computed geometric structures such as the Voronoi or Delaunay diagrams [9]. In this work, the desired diagram is reconstructed incrementally, adding the points in stages. In each stage, the algorithm faces a batched point location problem to add the current points to the diagram defined by all the previously inserted points. The algorithm could use standard point location or line-sweep methods, but this has a logarithmic cost per point. Suppose instead that the algorithm knew how to connect the points by a path or spanning tree that crosses the edges of the diagram only linearly many times. Then, once one of the points is located, the spanning structure could be traversed through the diagram and the remaining points located in linear time.

The construction of such spanning trees motivated the current investigation, in which we generalize the subdivision edges to allow other classes of geometric objects. Let $\mathcal{P}$ be a set of n points in the plane, which we call *sites,* and let $\mathcal{B}$ be a set of m geometric objects, which we call *barriers.* We assume that no site lies inside any of the barriers. An edge e, which is a straight line segment joining two sites, has a *cost* $c(e)$ that equals the number of barriers that e intersects. (When

[1]Japan Advanced Institute of Science and Technology, Asahidai, Tatsunokuchi, Ishikawa, 923-1292 Japan. Partly supported by Grant in Aid for Scientific Research of the Ministry of Education, Science and Cultures of Japan.

[2]Institute of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands.

[3]Department of Computer Science, Stanford University, Stanford, CA 94305 USA.

[4]Department of Computer Science, University of North Carolina at Chapel Hill, NC 27599–3175. Research partly supported by NSF grants 9988742 and 0076984. Much of this work was performed at the University of British Columbia, with support from National Science and Engineering Research Council of Canada.

[5]Department of Computer Science, Meiji University, Higashi-Mita, Tama-ku, Kawasaki, 214 Japan. Partly supported by Grant in Aid for Scientific Research of the Ministry of Education, Science and Cultures of Japan.

barriers are non-convex, an edge may intersect a barrier more than once. Such a barrier will still be counted only once.) The cost of a spanning tree $\mathcal{T}$ for $\mathcal{P}$ is the sum of the costs of its edges:

$$c(\mathcal{T}) = \sum_{e \in \mathcal{T}} c(e).$$

(It would be more precise to speak of the cost *with respect to* $\mathcal{B}$, but since the barrier set will always be fixed and clear from the context, we omit this addition.) We are interested in cheap spanning trees, that is, spanning trees with small cost, for several types of barriers. We obtain the following results.

Section 2 deals with the case in which the barriers are possibly intersecting line segments. Here we show that there are configurations in which any spanning tree has cost $\Omega(\min(m^2, m\sqrt{n}))$. We also show how to construct a spanning tree with this cost.

Section 3 deals with various types of disjoint barriers. Here it turns out that much cheaper spanning trees can be constructed. For instance, we are able to obtain a bound of $O(n+m)$ when the barriers are fat objects—discs for example. This bound is tight in the worst case.

The major result in this paper is given in Section 4, where we prove that for any set of n sites and any set of m barriers that are disjoint convex sets, there is a spanning tree of cost $O(m+n)$. This is optimal in the worst case. If the barriers are line segments, we show that there exists a spanning tree in which no barrier segment is crossed more than four times. Thus, the batched point location problem above can be solved by treating subdivision edges as barriers and forming such a linear cost spanning tree.

All our proofs are constructive. Our construction in Section 3 indeed leads to an efficient $O((n+m)\log m)$ algorithm to produce a spanning tree of low cost. The existence proofs are more interesting, however, since a simple greedy algorithm will always construct a spanning tree of minimal cost (and for the linear-time reconstruction goal the computation of the tree happens during the preprocessing in any case).

The bounds mentioned above are significantly better then the naive $O(nm)$ bound. We close this introduction by noting that if we wish to construct a triangulation on the sites, not just a spanning tree, then the naive bound is tight in the worst case. This can be seen by the example in Fig. 1.

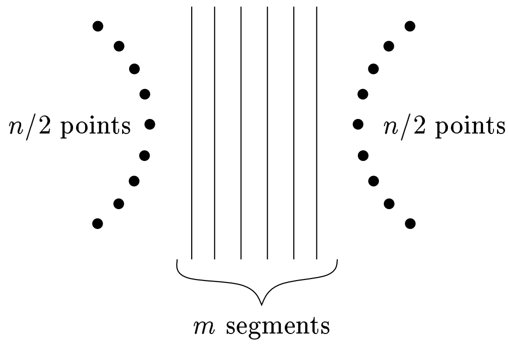


Figure 1: Any triangulation of the point set will have cost $\Omega(nm)$.

2 Intersecting segment barriers

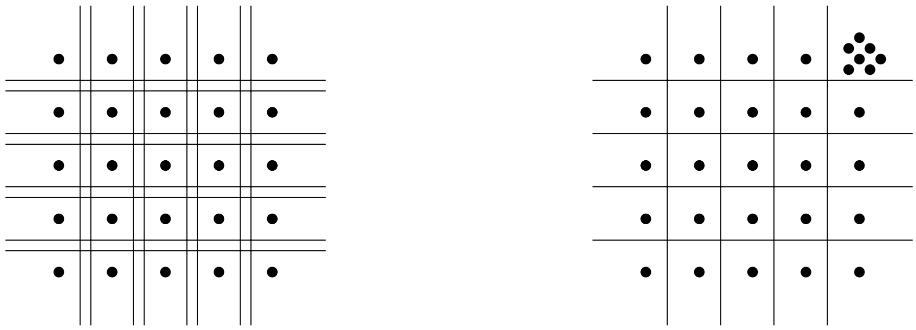
We start with the case of barriers in $\mathcal{B}$ that are possibly intersecting line segments.

Theorem 2.1 *(i) For any set $\mathcal{P}$ of n sites and any set $\mathcal{B}$ of m possibly intersecting segments in the plane, a spanning tree for $\mathcal{P}$ exists with a cost of $O(\min(m^2, m\sqrt{n}))$.*

(ii) For any n and m there is a set $\mathcal{P}$ of n sites and a set $\mathcal{B}$ of m segments in the plane, such that any spanning tree for $\mathcal{P}$ has a cost of $\Omega(\min(m^2, m\sqrt{n}))$.

Proof. (i) Extend the line segments in $\mathcal{B}$ to full lines. For each cell in the resulting arrangement, if the cell contains sites, choose a representative site and connect all sites in that cell to the representative. The edges used for this have zero cost, since the cells are convex and contain no barriers. Finally, compute a spanning tree on the set of representative sites with the property that any line intersects $O(\sqrt{n'})$ edges of the spanning tree [4], where n' is the number of representatives. The cost of the spanning tree is $O(m\sqrt{n'})$. This proves part (i) of the theorem, since $n' \leqslant \min(n, m^2)$.

(ii) First consider the case with $m \geqslant 2\sqrt{n} - 2$. We assume for simplicity that n is a square. We place the sites in a regular $\sqrt{n} \times \sqrt{n}$ grid. In between any two consecutive rows we put a bundle of $\lfloor m/(2\sqrt{n} - 2) \rfloor$ horizontal barrier segments; in between any two consecutive columns we put a bundle of $\lfloor m/(2\sqrt{n} - 2) \rfloor$ vertical segments. The remaining segments are placed arbitrarily. Figure 2(a) shows the construction for the case $n = 25$ and $m = 16$. Any



(a) construction for $m \geqslant 2\sqrt{n} - 2$ (b) construction for $m < 2\sqrt{n} - 2$

Figure 2: The lower bound constructions.

edge connecting two sites crosses at least one bundle. Hence, the cost of any spanning tree is at least $(n-1)\lfloor m/(2\sqrt{n} - 2) \rfloor = \Omega(m\sqrt{n})$.

Now consider the case with $m < 2\sqrt{n} - 2$. We arrange the barrier segments as shown in Fig. 2(b) for the case $m = 8$: we have a group of $\lfloor m/2 \rfloor$ vertical segments and a group of $\lceil m/2 \rceil$ horizontal segments, such that any vertical segment intersects any horizontal segment. We place a site in each of the resulting "cells"; the remaining sites are placed in any cell. Any spanning tree for $\mathcal{P}$ will have cost $\Omega(m^2)$.

□

3 Disjoint uncluttered barriers

Let $\mathcal{P}$ be a set of n sites in the plane, $\mathcal{B}$ a set of m disjoint barriers (of arbitrary shape). We give an algorithm that uses a *binary space partition* (BSP) for the set of barriers to construct a spanning tree for $\mathcal{P}$. We analyze the cost of the resulting spanning tree for orthogonal BSPs. Combining our analysis with known results on BSPs will then give us cheap spanning trees for so-called uncluttered scenes (defined below).

Given a BSP, our algorithm recursively associates with each node ν of the BSP a subset $\mathcal{P}_\nu \subset \mathcal{P}$, and constructs a spanning tree for $\mathcal{P}_\nu$. Initially ν is the root of the BSP and $\mathcal{P}_\nu = \mathcal{P}$; the final result is a spanning tree for $\mathcal{P}$. There are three cases to consider.

(i) If $\mathcal{P}_\nu$ contains at most one site, then we already have a spanning tree for $\mathcal{P}_\nu$.

(ii) If $\mathcal{P}_\nu$ contains more than one site but ν is a leaf of the BSP, then we connect the sites into a spanning tree in an arbitrary manner.

(iii) In the remaining case, $\mathcal{P}_\nu$ contains more than one site and ν is an internal node of the BSP. Let ℓ_ν be the splitting line stored at ν. The line ℓ_ν partitions $\mathcal{P}_\nu$ into two subsets. (Points on the splitting line all go to the same subset, say the right one.) We recursively construct a spanning tree for each of these subsets by visiting each child of ν with the relevant subset. Finally, if both subsets are non-empty we connect the two spanning subtrees by adding an edge between the sites closest to ℓ_ν on either side of ℓ_ν.

We now analyze the worst-case cost of the constructed spanning tree for the special case of *orthogonal BSPs*. (An orthonal BSP for $\mathcal{B}$ is a BSP whose splitting lines are all horizontal or vertical.) We assume that the leaves of the BSP store at most c objects, for some constant c; thus the cells of the final subdivision are intersected by at most c objects. (Note that we cannot require $c = 0$ unless we restricted our attention to orthogonal barrier segments.) The number of *fragments* generated by the BSP is the sum of the number of barriers stored at each leaf, over all leaves.

The following result will imply the existence of spanning trees of linear cost for several classes of barriers, including orthogonal segments and convex fat objects.

Theorem 3.1 *Let $\mathcal{B}$ be a set of disjoint simply-connected barriers in the plane, and let $\mathcal{P}$ be a set of n sites in the plane. Suppose an orthogonal BSP for $\mathcal{B}$ exists that generates f fragments and whose leaf cells intersect at most c barriers. Then there is a spanning tree for $\mathcal{P}$ with cost at most $O(f + k + cn)$, where k is the total number of vertical and horizontal tangencies on barrier boundaries.*

Proof. We consider the cost of the spanning-tree edges added in different cases of our algorithm. Each edge added in case (ii) intersects at most c barriers, so their total cost is at most $c(n-1)$.

Now consider an edge pq added in case (iii), and assume that the splitting line ℓ_ν is vertical. Let region(ν) denote the region corresponding to ν. Since the BSP uses only horizontal and vertical splitting lines, region(ν) is a rectangle, possibly unbounded on one or more sides. Define R_ν to be the intersection of region(ν) with the slab bounded by vertical lines through the sites p and q—see Fig. 3. By the choice of p and q, there are no other points in gap R_ν, so R_ν will not overlap with vertical slabs in the subtree of ν.

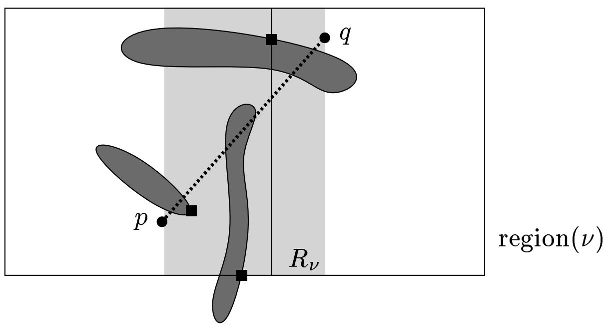


Figure 3: Illustration for the proof of Theorem 3.1.

Let b be a barrier intersected by pq. We will show how to charge this intersection to certain *features* of the barriers. These features are:

- The intersections between barrier boundaries and splitting lines. The number of these features is linear in the number of fragments f.
- Vertical and horizontal tangencies of barrier boundaries. The number of these features is k.

The charging of the intersection of pq with b is done as follows.

- If the boundary of b has a vertical tangent in the interior of R_ν, then we charge the intersection to this feature.
- Otherwise the boundary of b either intersects ℓ_ν in a point r lying in the interior of region(ν), or it intersects the boundary of region(ν) in a point r' that is also on the boundary of R_ν. Now we charge the intersection to r or r', respectively. Observe that both r and r' are features of b.

Fig. 3 shows, for each of three intersected barriers, a feature to which the intersection with pq can be charged. (Notice that there may be some choice, which can be made arbitrarily.) To bound the number of times a feature gets charged, we observe that the regions R_ν of nodes ν whose splitting line is vertical have disjoint interiors. It follows that a vertical tangency is charged at most once, and an intersection of a barrier boundary with a splitting line is charged at most twice (namely at most once for each fragment that has the point as a vertex). Similarly, a feature is charged at most twice from a node whose splitting line is horizontal. $\square$

A κ-cluttered scene in the plane is a set $\mathcal{B}$ of objects such that any square whose interior does not contain a bounding-box vertex of any of the the objects in $\mathcal{B}$ is intersected by at most κ objects in $\mathcal{B}$. A scene is called uncluttered if it is κ-cluttered for a (small) constant κ. It is known that any set of disjoint fat objects, discs for instance, is uncluttered—see the paper by de Berg et al. [3] for an overview of these models and the relations between them.

Theorem 3.2 *Let $\mathcal{B}$ be a set of m disjoint objects in the plane, each with a constant number of vertical and horizontal tangents, that form a κ-cluttered scene, for a (small) constant κ. Let $\mathcal{P}$ be a set of n sites. Then there is a spanning tree for $\mathcal{P}$ with cost $O(m+n)$. This bound is tight in the worst case, even for unit discs. A spanning tree with this cost can be computed in time $O((m+n)\log m)$.*

Proof. De Berg [2] has shown that a κ-cluttered scene admits an orthogonal BSP that generates $O(m)$ fragments such that any leaf cell of the BSP is intersected by at most $O(\kappa)$ fragments. Then by Theorem 3.1 there is a spanning tree of cost $O(m + \kappa n)$.

To see that this bound is tight, take a disc as the only barrier and place the sites around the disc and so close to it that any edge connecting two sites crosses the disc. In this situation any spanning tree must have cost $\Omega(n)$. A row of m discs with two sites on either side is an example in which any spanning tree must have cost $\Omega(m)$.

De Berg gives an algorithm that constructs the orthogonal BSP in time $O(m \log m)$, given only the corners of the bounding boxes of the barriers. The BSP induces a planar subdivision consisting of $O(m)$ boxes. We assign each site to the box containing it in time $O(n \log m)$ [2], and then construct the spanning tree from the leaves of the BSP upwards. Since we only need to maintain the leftmost, rightmost, topmost, and bottommost site in each node of the BSP, this can be done in time $O(n+m)$. $\square$

Theorem 3.1 also implies that we can always find a spanning tree of cost $O(m)$ when the barriers are disjoint orthogonal segments, because Paterson and Yao [7] have shown that any set of orthogonal line segments in the plane admits an orthogonal BSP of size $O(m)$ whose leaf cells are empty. We can construct such a spanning tree in time $O((n+m)\log m)$: we need $O(m\log m)$ time to construct the BSP [5], plus $O((n+m)\log m)$ time to locate the sites in the BSP subdivision using an optimal point location structure [6], and $O(n+m)$ for the bottom-up construction of the spanning tree.

In the next section we will show that a linear-cost spanning tree exists for any set of disjoint barrier segments (even if they are not orthogonal), however, we do not know of an equally efficient way to construct the tree in the general case.

4 Disjoint convex barriers

We now present the main result of our paper: Given any set $\mathcal{P}$ of n sites in the plane and any set $\mathcal{B}$ of m disjoint convex barriers that do not contain any sites, there is a spanning tree of $\mathcal{P}$ whose cost is at most $4m + 3n$. When the barriers are line segments, we can improve the upper bound to $4m$ by ensuring that no segment barrier is intersected more than 4 times. We will concentrate primarily on the case of segment barriers in our illustrations and examples.

One can obtain a spanning tree of cost $O(m\log(n+m))$ for segments in several ways. One way is to analyze a slightly adapted version of the BSP-based algorithm in terms of the depth of the underlying BSP, and use the fact that any set of m disjoint segments in the plane allows a BSP of size $O(m\log m)$ and depth $O(\log m)$ [7]. Another way is to use a divide-and-conquer approach based on cuttings. With neither of these two approaches have we been able to obtain a linear bound. The solution presented next, therefore, uses a different, incremental approach.

We assume that the barriers and the sites are all *strictly* contained in a fixed bounding box, say an axis parallel unit square. We denote the upper-left and the upper-right corners of the bounding box by c_l and c_r, respectively. We will assume in the following that the sites, the common tangents of barriers in $\mathcal{B}$, and the two points c_l and c_r are in general position collectively. This is not a serious restriction, but does make the description easier.

Let $\mathcal{T}$ be a spanning tree on $\mathcal{P} \cup \{c_l, c_r\}$ with straight edges and no self-intersections. (Note that the minimum cost spanning tree may require self-intersections. Our construction proves that self-intersections are not necessary to achieve the linear bound.) For two sites q, r of $\mathcal{T}$, we denote by $path(q, r)$ the path between q and r in $\mathcal{T}$.

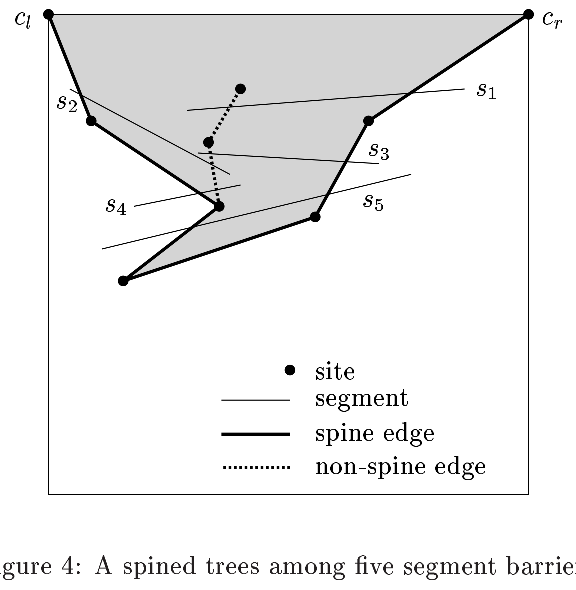


Figure 4: A spined trees among five segment barriers.

We call $path(c_l, c_r)$ the *spine* of $\mathcal{T}$. The spine of $\mathcal{T}$ partitions the bounding box into two parts: the part *above the spine* which is bordered by $path(c_l, c_r)$ and the upper edge of the bounding box, and the remaining part *below the spine*. Note that a point above the spine, in this definition, may see some edge of the spine above it since we do not assume x-monotonicity of $path(c_l, c_r)$. We say that the tree $\mathcal{T}$ is *spined* if

(1) all the sites are either on or above the spine, and
(2) both c_l and c_r are leaves of $\mathcal{T}$.

Fig. 4 shows an example of a spined tree among 5 segment barriers and 9 sites, including the artificial sites c_l and c_r. The spine of the spined tree $\mathcal{T}$ is depicted by solid bold lines.

We will show how to build a spined tree by inserting sites in order of decreasing y-coordinate, starting with the single edge $c_l c_r$. The construction of the tree will be the same for all disjoint, convex barriers; the inductive analysis will be different for segment barriers and for general convex sets. Before we begin, we need some additional notation and lemmas for barriers and their interaction with a spined tree $\mathcal{T}$.

The spine of $\mathcal{T}$ may intersect a barrier b several times, cutting b into a number of connected components. We call the connected components lying *below the spine* the *barrier components* of b. Recall that sites are not allowed to lie inside barriers, so each intersection of the spine with b is an interval on a single spine edge, and the barrier components are convex. The intersections with the spine incident on a barrier component are called its *anchors*.

The next definitions are made with respect to a chosen point p in the bounding box below the spine of $\mathcal{T}$. As p will be fixed throughout the following arguments, our notation does not show the dependence on p.

The point p induces a depth-order on the barriers (more precisely, on the parts of the barriers lying above the horizontal line through p). We say that a barrier b_2 *obscures* b_1, and write $b_2 \prec b_1$, if a ray with positive y-direction starting at p intersects b_2 before it intersects b_1, as in Fig. 5. The relationship $\prec$ is acyclic, and its transitive closure is a partial order [1, Section 10.5].

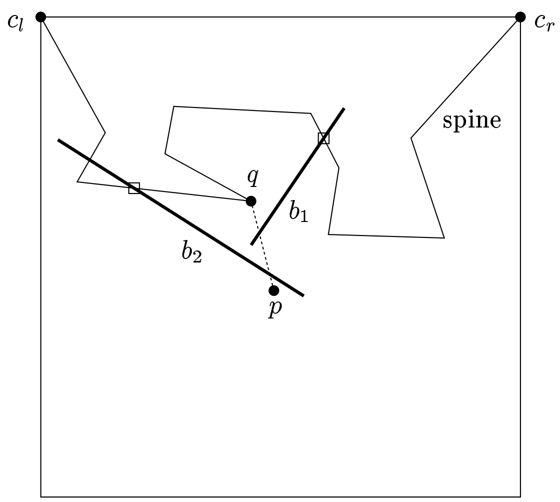


Figure 5: We have $b_2 \prec b_1$. Two of the three barrier components below the spine block q. Their anchors are marked by squares.

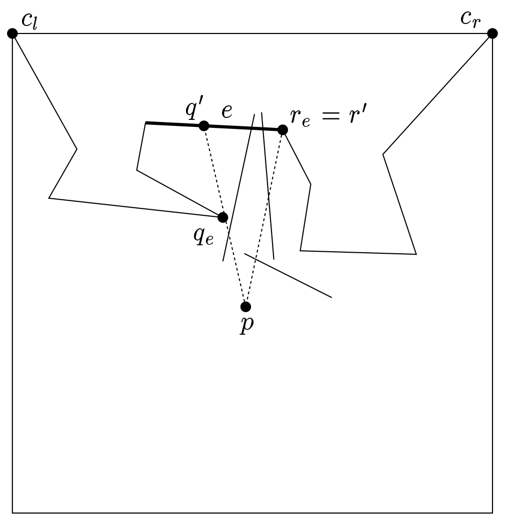


Figure 6: A good edge e, with the barriers that block points q_e and r_e.

We say that a point q in the bounding box is *visible from* p if the segment pq does not intersect the spine of $\mathcal{T}$ except possibly at q. (Barriers do not play a role in this definition.) We will call an edge e of the spine *visible from* p if it contains points visible from p in its interior.

Let $Q = \{q_1, q_2, \dots, q_t\}$ denote the set of sites of the spine of $\mathcal{T}$ that are visible from p, listed in order from c_l to c_r. Note that $q_1 = c_l$ if c_l is visible from p and $q_t = c_r$ if c_r is visible.

For a spine edge e, we define q_e to be the first site visible from p when we traverse the spine of $\mathcal{T}$ from e towards c_l. We define r_e similarly as the first site when we traverse from e to c_r. The sites q_e and r_e are consecutive in the sequence of visible sites Q.

Let q', r' be the points on $path(q_e, r_e)$ such that $path(q', r')$ is the maximal subpath of $path(q_e, r_e)$ that is visible from p. Note that p, q', q_e are collinear with possibly $q_e = q'$ and p, r', r_e are collinear with possibly $r_e = r'$, as in Fig. 6. If the edge e is visible from p, then e is the only visible edge on $path(q_e, r_e)$, and the points q', r' lie on e.

For a point q on the spine visible from p, we say that a barrier component b *blocks* q if it intersects the line segment pq. The following proposition will be used later.

Proposition 4.1 *For a spine edge e, a barrier component that blocks q_e has no anchor in $path(q_e, q')$, and a barrier component that blocks r_e has no anchor in $path(r', r_e)$.*

From the perspective of point p, an edge e of the spine is called a *good edge* if and only if

(1) edge e is visible from p,
(2) any anchors left of r_e for barrier components that block r_e are on e,
(3) any anchors right of q_e for barrier components that block q_e are on e.

We will see that a good edge can be used to extend a spined tree $\mathcal{T}$ down to p by adding segments pq_e and pr_e and deleting edge e.

Lemma 4.2 *For a point p in the bounding box that has smaller y-coordinate than any site of the spined tree $\mathcal{T}$, there is at least one good edge e.*

Proof. Let B denote the set of barrier components that have anchors on the spine and that block sites on the spine. We assume that B is non-empty, as otherwise the existence of a good edge is trivial.

Choose barrier component $b_0 \in B$ that is minimal with respect to the "obscures" relation $\prec$, and let e be a spine edge containing an anchor of b_0. Let $q = q_e$, $r = r_e$. Since b_0 blocks a site on the spine and has an anchor on e, it blocks at least one of q and r.

We show that either e is a good edge, or there is a good edge adjacent to q or r. We consider cases that depend on whether the anchor edge e is visible from p.

Case 1: Suppose that the anchor edge e is visible from p—equivalently, e contains the points q' and r'. By minimality of b_0, no other $b \in B$ can intersect both pq and pr. By Proposition 4.1, therefore, any barrier component that blocks q must have its anchor left of q or right of q' on e. Similarly, any barrier component that blocks r must have its anchor right of r or left of r' on e. Thus, e is a good edge.

Case 2: Suppose that e is not visible, and assume, without loss of generality, that b_0 blocks q. Note that b_0 cannot also block r because its anchor at e is on $path(r', r)$. In fact, no barrier component $b \in B$ can block r because of the minimality of b_0 along pr'. But now the edge f that is immediately to the right of r along the spine must be visible from p. Let $w = r_f$ be the first visible site right of r along the spine. Any barrier component that blocks w and has an anchor left of w must therefore have this anchor on f. Thus, f is a good edge.

In either case we find a good edge, and the lemma is proven. $\square$

Using a good edge, we can construct a spined tree of $\mathcal{P} \cup \{c_l, c_r\}$. We do so first for segment barriers.

Lemma 4.3 *Given a bounding box, with upper corners c_l and c_r, that contains $\mathcal{P}$, a set of n sites, and $\mathcal{B}$, a set of m barriers that are disjoint line segments, then there is a spined tree $\mathcal{T}$ of $\mathcal{P} \cup \{c_l, c_r\}$ such that each segment $b \in \mathcal{B}$ is stabbed by $\mathcal{T}$ at most $2 + u(b)$ times, where $u(b)$ denotes the number of endpoints of b that are above the spine of $\mathcal{T}$ (and hence is at most two).*

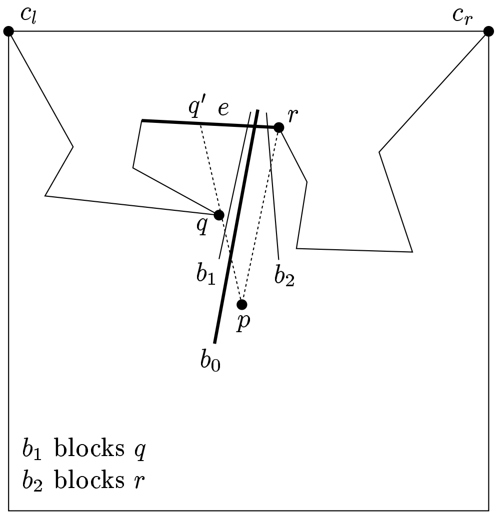


Figure 7: Case 1

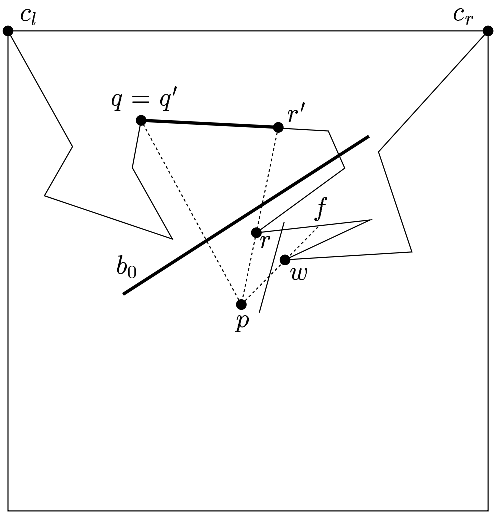


Figure 8: Case 2.

Proof. The proof is by induction on the number of sites in $\mathcal{P}$. We fix the barrier set $\mathcal{B}$ throughout.

If $\mathcal{P}$ is empty, there are only two sites c_l and c_r. The edge between c_l and c_r does not stab any barrier of $\mathcal{B}$, so the claim holds.

Assume now that $\mathcal{P}$ contains at least one site. Let p be the lowest site, that is, the site with the smallest y-coordinate, and let $\mathcal{P}' = \mathcal{P} \setminus \{p\}$. Let $\mathcal{T}'$ be the spined tree of $\mathcal{P}' \cup \{c_l, c_r\}$ provided by the inductive assumption.

By Lemma 4.2, the spine of $\mathcal{T}'$ contains a good edge e. Let $q = q_e$, $r = r_e$. Our spined tree $\mathcal{T}$ is obtained from $\mathcal{T}'$ by adding the two edges pq and pr and removing e. Since q and r are visible from p, we do not create any self-intersections, and since e is in $path(q, r)$, $\mathcal{T}$ remains a tree. The new spine goes through the edges pq and pr and it is clear that all sites are either on or above this spine. If $q = c_l$ (or $r = c_r$) then the visible edge e must be incident on c_l (or c_r), which guarantees that c_l and c_r remain leaves of the tree. Therefore, $\mathcal{T}$ is indeed a spined tree of $\mathcal{P} \cup \{c_r, c_r\}$.

New stabbings are created when a barrier segment $b \in \mathcal{B}$ is stabbed by pq or pr. We have three subcases: (a) b is stabbed by both pq and pr, (b) b is stabbed by pq but not by pr, and (c) b is stabbed by pr but not by pq. Since (c) is symmetric to (b), we consider subcases (a) and (b).

In case (a), we first observe that the barrier components of b blocking q and r are identical, as the triangle pqr does not intersect the spine of $\mathcal{T}'$. It follows that this barrier component has no anchor on any edge of $\mathcal{T}'$ because conditions (2) and (3) of a good triple imply that the only anchor edge could be e, which is not possible for the line segment b. Thus, the stabbing number of b becomes two without violating the inductive assumption.

In case (b), b is stabbed by pq but not by pr. Let C denote the closed curve formed by edges pq, pr and $path(q, r)$. Let b' be the barrier component of b blocking q.

First suppose that b' has no anchor on $path(q, r)$. Then one endpoint of b is in the interior of the cycle C. Since the interior of C is below the spine of $\mathcal{T}'$ and above the spine of $\mathcal{T}$, the number of endpoints of b above the spine is increased by one, accounting for the new stabbing and maintaining the inductive assumption.

Next suppose that b' has an anchor on $path(q, r)$. By Proposition 4.1 and the conditions of a good triple, this anchor must lie on e. Since e is removed in forming $\mathcal{T}$, the inductive assumption is maintained in this case as well. $\square$

We now present our first main theorem.

Theorem 4.4 *Given a set $\mathcal{B}$ of non-intersecting line segments and a set $\mathcal{P}$ of sites in the plane,* *there is always a straight-edge spanning tree of $\mathcal{P}$ without self-intersections that stabs each line segment of $\mathcal{B}$ at most 4 times.*

Proof. We choose a bounding box that properly contains all the objects of $\mathcal{B}$ and $\mathcal{P}$. Let c_l and c_r be the upper-left and upper-right corners of the bounding box, respectively. Then, applying Lemma 4.3 to $\mathcal{B}$ and $\mathcal{P} \cup \{c_l, c_r\}$, we obtain a spined tree $\mathcal{T}$. The artificial sites c_l and c_r are leaves of $\mathcal{T}$, and removing them results in a spanning tree of $\mathcal{P}$ that stabs each line segment of $\mathcal{B}$ at most 4 times. □

We now turn to the case of convex barriers. We can prove a simple bound using Theorem 4.4 as follows.

Corollary 4.5 *Given a set $\mathcal{B}$ of m non-intersecting convex barriers and a set $\mathcal{P}$ of n sites in the plane that are not contained in any barriers, there is always a straight-edge spanning tree of $\mathcal{P}$ without self-intersections with cost at most $n + O(m)$.*

Proof. Construct a straight-edge planar subdivision of complexity $O(m)$, such that each barrier is completely contained in the interior of a face, and no face contains more than one barrier [1]. Let $\mathcal{B}'$ be the set of edges of the subdivision. Applying Theorem 4.4 on $\mathcal{P}$ and $\mathcal{B}'$ results in a spanning tree $\mathcal{T}$ of $\mathcal{P}$ without self-intersections that intersects $\mathcal{B}'$ at most $O(m)$ times. If an edge e of $\mathcal{T}$ intersects k segments of $\mathcal{B}'$, it can intersect at most $k+1$ faces of the subdivision, and therefore at most $k+1$ barriers in $\mathcal{B}$. If follows that the cost of $\mathcal{T}$ with respect to $\mathcal{B}$ is at most $n+O(m)$. □

We can establish a better dependence on m if we refine the analysis of our spined tree.

Lemma 4.6 *Given a bounding box, with upper corners c_l and c_r, that contains $\mathcal{P}$, a set of n sites, and $\mathcal{B}$, a set of m barriers that are disjoint convex sets, then there is a spined tree $\mathcal{T}$ of $\mathcal{P} \cup \{c_l, c_r\}$ such that*

$$z_1 + 2m_1 + n_1 + 2m_2 \leqslant 3n + 4m,$$

where z_1 is the number of intersections of barriers with non-spine edges of $\mathcal{T}$, m_1 is the total number of barrier components, n_1 is the number of sites on the spine, and m_2 is the number of barriers lying stricly below the spine.

Proof. The inductive construction of the tree is identical to that in the proof of Lemma 4.3. We therefore concentrate on maintaining the inductive assumption as we add p to tree $\mathcal{T}'$ to form $\mathcal{T}$. That is, as we take a good edge e, let $q = q_e$ and $r = r_e$, add pq and pr, and delete edge e. Let z_1', m_1', m_2', n_1' denote the quantities of the inductive assumption for $\mathcal{T}'$.

If ℓ is the number of vertices on $path(q,r)$, not counting q and r themselves, then $n_1 = n_1' + 1 - \ell$. Let C denote the closed curve formed by pq, pr, and $path(q,r)$, and let k_1 be the number of barrier components lying inside C. Since e is a good edge, a barrier component that has an anchor on $path(q,r)$ outside of e cannot intersect either pq or pr, and so all such intersections are created by the k_1 barrier components in C. We count these intersections using the following Davenport-Schinzel sequence [8]: Label each of the k_1 barrier components, then walk along $path(q,r)$ and create the sequence of labels encountered. For any consecutive identical labels, omit all but one representative. The resulting sequence cannot have an *abab* subsequence, since that would indicate that barrier components a and b intersect below the spine. The longest such sequence with k_1 letters has length $2k_1 - 1$. Each repetition can be charged against a vertex of $path(q,r)$, since each edge on this path can intersect a convex barrier component at most once. Thus, the total number of intersections is at most $2k_1 + \ell$, and so $z_1 \leqslant z_1' + 2k_1 + \ell$.

It remains to bound the increase in the total number of barrier components. The number of components of a barrier b can only increase if a component of b blocks both q and r. Let k_2 be the number of such barrier components. We have $m_1 = m_1' - k_1 + k_2$. Conditions (2) and (3) of a good edge imply that a barrier component blocking both q and r cannot have an intersection with the spine of $\mathcal{T}'$, except possibly in e. Furthermore, among these k_2 barrier components, only one can intersect e. This implies that $m_2 \leqslant m_2' - (k_2 - 1)$.

To summarize, we have

$$\begin{aligned} z_1 + 2m_1 + n_1 + 2m_2 &\leqslant (z_1' + 2k_1 + \ell) + 2(m_1' - k_1 + k_2) + (n_1' + 1 - \ell) + 2(m_2' - k_2 + 1) \\ &= z_1' + 2m_1' + n_1' + 2m_2' + 3 \leqslant 3(n-1) + 4m + 3 = 3n + 4m, \end{aligned}$$

which completes the proof. □

Theorem 4.7 *Given a set $\mathcal{B}$ of m non-intersecting convex barriers and a set $\mathcal{P}$ of n sites in the plane that are not contained in any barriers, there is always a straight-edge spanning tree of $\mathcal{P}$ without self-intersections with cost at most $4m + 3n$.*

Proof. As in the previous proof, construct a bounding box with upper corners c_l and c_r. Applying Lemma 4.6 to $\mathcal{B}$ and $\mathcal{P} \cup \{c_l, c_r\}$, we obtain a spined tree $\mathcal{T}$. Let z_2 be the number of intersections between barriers and the spine edges of $\mathcal{T}$. Using the Davenport-Schinzel argument of the lemma, we find that

$$z_2 \leqslant 2m_1 + n_1,$$

and so the total number of intersections between barriers and the tree $\mathcal{T}$ is bounded by

$$z_1 + z_2 \leqslant z_1 + 2m_1 + n_1 + 2m_2 \leqslant 3n + 4m.$$

Removing the artificial sites c_l and c_r from $\mathcal{T}$ gives the final spanning tree without increasing this cost. □

When n is large compared to m, we can prove a better bound—but the tree may become self-intersecting.

Corollary 4.8 *Given a set $\mathcal{B}$ of m non-intersecting convex barriers and a set $\mathcal{P}$ of n sites in the plane that are not contained in any barriers, there is always a straight-edge spanning tree of $\mathcal{P}$ (possibly with self-intersections) with cost at most $13m + n + 3$.*

Proof. We construct a vertical decomposition for $\mathcal{B}$. This is a partition of the complement of $\bigcup \mathcal{B}$ into at most $3m + 1$ "trapezoids"—regions bounded from above and below by the boundary of one barrier, and on the left and right by vertical segments. Each trapezoid can be split by a line segment into two "sub-trapezoids" bounded on one side by a barrier and on the other three sides by straight segments.

Let now $\mathcal{P}_1$ be a set of representative points obtained by chosing one arbitrary site of $\mathcal{P}$ in each trapezoid that contains sites. Furthermore, let $\mathcal{P}_2$ be another point set obtained by chosing one arbitrary site of $\mathcal{P}$ in each sub-trapezoid that contains at least one site but no point of $\mathcal{P}_1$. Let k_1 and k_2 be the cardinality of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively.

We now apply Theorem 4.7 to the sets $\mathcal{P}_1$ and $\mathcal{B}$, resulting in a spanning tree $\mathcal{T}$ of $\mathcal{P}_1$, with cost at most $4m + 3k_1$. We then connect each point in $\mathcal{P}_2$ to the representative point in the same trapezoid. This results in at most $2k_2$ intersections with barriers (namely at most two in each trapezoid). Finally, we connect each point in $\mathcal{P} \setminus (\mathcal{P}_1 \cup \mathcal{P}_2)$ to the point of $\mathcal{P}_1 \cup \mathcal{P}_2$ in the same sub-trapezoid. This results in at most $n - (k_1 + k_2)$ intersections (namely at most one in each sub-trapezoid). The cost of the resulting tree is therefore $4m + 3k_1 + 2k_2 + n - k_1 - k_2 = 4m + 2k_1 + k_2 + n$. Since k_1 and k_2 are bounded by $3m + 1$, the claim follows. □

5 Conclusions

In this paper we have studied spanning trees among n points whose edges cross few among a given set of m barriers. When the barriers are disjoint, near-linear bounds for the cost of such a tree can be obtained by several simple arguments. Using more sophisticated techniques, we were able to show that a linear-cost spanning tree is possible in many cases.

We note that the number of barriers crossed by linking two points is not a distance function and does not satisfy the triangle inequality. This means that the existence of other low-cost structures among the points, such as Hamiltonian paths and matchings, remains an interesting research problem.

While we have given an efficient algorithm to compute spanning trees based on an orthogonal BSP, we are not aware of an equally efficient algorithm to construct a linear-cost spanning tree for the case of arbitrary line segments or convex barriers. Can this be done in $O(n \log n)$ time?

The constants in our bounds of Section 4 are not tight. It would be especially interesting to show a tight bound on the maximum number of crossings of segment barriers. Is there a configuration that requires four crossings for at least one barrier?

References

[1] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal. Lecture Notes Comput. Sci.*, vol. 703. Springer-Verlag, 1993.

[2] M. de Berg. Linear size binary space partitions for uncluttered scenes. *Algorithmica*, 28:353–366, 2000.

[3] M. de Berg, M. Katz, F. van der Stappen, and J. Vleugels. Realistic input models for geometric algorithms. *Algorithmica*. To appear.

[4] B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.*, 4:467–489, 1989.

[5] F. d'Amore and P. G. Franciosa. On the optimal binary plane partition for sets of isothetic rectangles. *Inform. Process. Lett.*, 44:255–259, 1992.

[6] H. Edelsbrunner, L. J. Guibas, and J. Stolfi. Optimal point location in a monotone subdivision. *SIAM J. Comput.*, 15:317–340, 1986.

[7] M. S. Paterson and F. F. Yao. Efficient binary space partitions for hidden-surface removal and solid modeling. *Discrete Comput. Geom.*, 5:485–503, 1990.

[8] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, New York, 1995.

[9] J. Snoeyink and M. van Kreveld. Linear-time reconstruction of Delaunay triangulations with applications. In *Proc. Annu. European Sympos. Algorithms*, number 1284 in Lecture Notes Comput. Sci., pages 459–471. Springer-Verlag, 1997.