

# TUFTS-CS Technical Report 03–06

June 2003

## Upper and Lower Bounds for Connecting Sites Across Barriers<sup>1</sup>

by

David Krumme <sup>2</sup>    Greg Perkins <sup>3</sup>    Eynat Rafalin <sup>†</sup>    Diane Souvaine <sup>†</sup>

---

<sup>1</sup>Research supported in part by NSF #EIA-99-96237

<sup>2</sup>Department of Computer Science, Tufts University, Medford, MA 02155. {dwk, erafalin, dls}@cs.tufts.edu

<sup>3</sup>Panasonic Technologies, Princeton, NJ 08540. gmp@research.panasonic.com

## ABSTRACT

Given a set  $M$  of  $m$  distinct points (*sites*) and a set  $N$  of  $n$  segments (*barriers*) in the plane, we address the problem of constructing a planar spanning tree of  $M$  of straight edges with minimal cost, where the cost function is the number of times that edges cross barriers (the *crossing number*). For the restricted problem where barriers form a convex planar subdivision and each face contains exactly one site, we prove a tight bound of  $\frac{5}{3}n$  total cuts. As for the number of cuts per barrier, we prove that at most 3 cuts per barrier are needed, provide a lower bound of 2 cuts per barrier and develop a linear time algorithm. Asano *et al.* [1] proved upper bounds of 4 cuts per barrier and a total cost of  $4n$  for the general case, with corresponding lower bounds of 2 and  $2n - 2$ . New constructions developed here may apply both to other problems and to tightening the bounds in the general case.

# 1 Introduction

In any geometric modeling system (e.g. robotics, vision, radio wave propagation prediction, CAD/CAM, ...), the question of where a set of points lie within the underlying geometric data structure arises. This pervasive problem in computer science is termed the *multi-point location problem*: given an  $O(n)$ -sized data structure  $P(E^p, V^p)$  which subdivides the plane (e.g. a triangulation) and a set  $M$  of  $m$  distinct points, locate for each  $p_i \in M$  the face that contains  $p_i$ , where  $i = 1, \dots, m$ . Through iterated use of one of the worst-case optimal planar point location algorithms (e.g. [5], [8] or [10]), one can locate all faces in  $O(m \log n)$  time. It remains an open problem to determine whether the faces can all be located in  $O(m+n)$  time. A potential strategy for locating all  $m$  points in  $O(m+n)$  time is to find a combinatorially  $O(m+n)$ -sized walk  $w$  that visits all  $m$  points<sup>1</sup>. A preorder traversal of a spanning tree  $T(E^t, M)$  with  $cost(T) = |E^t \cap E^p| \in O(n)$  would provide such a walk at a cost of  $2 cost(T)$ . We consider a specific version of this problem, posed by Snoeyink [11, 12]:

PROBLEM: Given a set  $\hat{M}$  of  $m$  distinct sites and a set  $\hat{N}$  of  $n$  barriers (line segments) in the plane where the relative interiors of all barriers are disjoint and no  $p_i \in \hat{M}$  lies on any  $s_j \in \hat{N}$ , does there always exist a spanning tree  $T$  of  $\hat{M}$  that, when embedded with straight edges, has the property that no segment in  $\hat{N}$  is cut by more than a constant number of cuts per edges?

Recent work by Asano *et al.* [1] demonstrated a lower bound of at least 2 cuts per edge (total of  $2n - 2$  cuts) and an upper bound of 4 cuts per edge (total of  $4n$ ). Hoffman and Toth [7] constructed a specific example with a lower bound of at most 3 cuts per edge.

In this paper we address the restricted problem where the set of barriers  $N$  forms a planar subdivision of convex faces, where the relative interiors of all barriers are disjoint. Every cell of the subdivision contains exactly one site of the extended set  $M$ . For simplicity and clarity we add 4 additional barriers that form a rectangle  $R$  such that all sites and all intersection points of barriers are contained inside  $R$ . For each infinite barrier that is crossed by the rectangle, we cut it at its intersection point with one of the barriers of the rectangle. In this problem  $|M| = |N| - 3$ .

We prove that all sites can be connected with straight edges (*connectors*) such that no barrier in  $N$  is cut by more than three connectors and that the total cost is at most  $\frac{5}{3}n$ . The lower bound for the restricted case remains 2 cuts per edge, but decreases to a total cost of  $\frac{5}{3}n$  (Figure 4 (a)).

The contribution of this paper also includes new constructions used in the proofs that should apply to other problems. The planar subdivision presented here corresponds to certain instances of binary space partitions [2, 3, 9, 14, 15]. Similar, but more restricted, constructions were used in the past in the study of binary space partitions [14, 15], and multi-way space partitioning trees [4] but only in a local setting, ignoring the global structure of the planar subdivision.

---

<sup>1</sup>This strategy was motivated by Snoeyink and Van Kreveld [13] who demonstrated empirically that walking to the next point to be processed with the method of Guibas and Stolfi [6] during the reconstruction of a triangulation  $T$  is better than traversing the faces of  $T$  as defined by their computed permutation.

Section 2 describes the problem. Section 3 introduces a greedy algorithm that is the first step in our construction and produces a *planar overlay map*. Section 4 defines a formal structure called a *separation diagram* that helps us prove important properties of the *planar overlay map*. Sections 5 and 6 prove some basic lemmas used in the main proofs in Sections 7 and 8. Section 9 describes an optimal algorithm and 10 describes an implementation. Section 11 provides summary and future work.

## 2 Problem Description

We transform each instance  $\hat{N}, \hat{M}$  of the original problem to an instance  $N, M$  of a restricted problem in which the set of barriers  $N$  forms a planar subdivision of convex faces where the relative interiors of all barriers are disjoint.

This planar subdivision presents a worst-case instance of the original problem (since the number of cuts can only increase by extending the barriers). We further restrict the problem, by adding additional barriers and Steiner points, such that every cell of the subdivision contains exactly one site of the extended set  $M$ . This is accomplished by recursively dividing each face that contains more than one site into several convex faces each containing one site, and by adding a new site inside each face that contains no site.

**Where a *connector* is an edge that connects two sites, our goal is to build a planar graph  $T$  of *connectors* that will span all the sites of  $M$  with minimal cost. The cost function is defined as the maximal number of times that any barrier will be crossed by connectors.**

## 3 A Greedy algorithm that almost solves the problem

The first step in building  $T$  is a *greedy algorithm* that adds connectors between sites such that every connector crosses only one barrier and every barrier is crossed at most once. Our algorithm builds such a maximal set of connectors. Any additional connector added to  $T$  in order to connect disjoint parts of the graph will have to cut a barrier a second time (or more). In order to verify that no barrier is crossed more than once, the algorithm maintains a set  $E$  of *excluded barriers* that have already been crossed once.

**The algorithm:** Let  $\hat{T}$  be the an initially empty set of planar connectors and  $\hat{E}$  an initially empty set of excluded segments. For each cell  $c_i$  of the subdivision, containing a site  $s_i \in M$ , and for each site  $s_j \in M$  in a cell adjacent to  $c_i$ : if edge  $(s_i, s_j)$  crosses only one barrier and does not cross any excluded barrier, add  $(s_i, s_j)$  to the set  $\hat{T}$ ; Add the barrier  $b_i$  crossed by  $(s_i, s_j)$  to the set  $\hat{E}$ . Continue as long as connectors can be added Let  $T$  be the maximal  $\hat{T}$ .

**Definition 3.1** *A planar overlay map is a triplet  $(M, N, T)$  of sites, barriers and connectors, each as defined above.*

This algorithm produces a graph  $G = (M, T)$  whose edges are connectors and vertices are sites which may or may not be connected. Define  $M_1, M_2, \dots, M_k$  to be the connected components of  $M$  created from the sites that are connected by edges of  $T$ .  $\bigcup_{i=1}^k M_i = M$  and  $M_i \cap M_j = \emptyset, \forall i \neq j$ .

For each  $M_i$ , define  $S_i$  to be the union of all the cells that contain sites of  $M_i$ , where a cell includes the segments of barriers that bound it (i.e. the closed cells of  $M_i$ ). Call the

components  $S_1, S_2, \dots, S_k$  sections. A connector crosses exactly one barrier and hence cannot connect two sites that are in cells that are not adjacent. Therefore, the sections will also be connected components and the boundary of each section is a simple polygon (with holes).  $\bigcup_{i=1}^k S_k$  covers the whole planar subdivision  $H$ . In addition, the intersection of two sections is a (possibly empty) collection of segments of barriers:  $S_i \cap S_j \subseteq N, \forall i \neq j$ . Call sections that share a barrier *adjacent sections*.

Clearly, to connect all sites, we need to build connectors that connect sites in different sections. Our task is to make sure that these connectors will not increase the crossing number above 3.

## 4 Separation Diagrams

In the following sections we investigate the relationships between adjacent sections. This is based on an analysis of the very special conditions that must pertain at a boundary between sections. We introduce an abstract arrangement of segments and points called a *separation diagram* which will serve as a kind of “straw man.” When some portion of a planar overlay map purports to be a boundary between sections, we argue that its structure can be matched to some separation diagram, which then implies that it cannot be such a boundary, because the greedy algorithm always bridges separation diagrams.

A separation diagram has three essential parts. First is a *backbone* which match a purported boundary between sections. Second are *excluded rays* which match obstacles that lie partly within the backbone and partly outside it. (The excluded ray matches the portion that lies outside the backbone.) Such obstacles are troublesome because the greedy algorithm may produce connectors that cross them away from the backbone, precluding their being crossed by any connector spanning the backbone. Third are *walls* at each end of the backbone which match obstacles with certain colinearity constraints. It turns out that the presence or absence of walls is the determining factor in whether the greedy algorithm will bridge the backbone and that absence of walls implies a cyclic structure for section boundaries.

Lemma 4.2 shows how a separation diagram is matched up to a planar overlay map. Lemma 4.5 proves that the greedy algorithm will never produce a boundary that under Lemma 4.2 can be matched up with a separation diagram. The consequence, roughly, is that the section boundaries match things that could be backbones of separation diagrams except that there are no walls.

**Definition 4.1** A separation diagram  $\Sigma$  (Figure 1) is a 6-tuple  $\{Q, C, \Lambda, \Xi, L, R, E\}$  where:

1.  $Q = \{q_0, q_1, \dots, q_s\}$  is a set of points of which  $q_1, q_2, \dots, q_{s-1}$  must all be distinct.
2.  $C$  (the backbone) is the piecewise linear curve  $\bigcup_{1 \leq i \leq s} [q_{i-1}, q_i]$ .
3.  $C$  has no self-intersections, unless  $q_0$  and/or  $q_s$  coincide with another member of  $Q$ .
4.  $\Lambda$  and  $\Xi$  are two sets of rays (half-lines)  $\Lambda = \{l_0, l_1, \dots, l_n\}$  and  $\Xi = \{r_0, r_1, \dots, r_m\}$  with endpoints  $p(l_0), \dots, p(l_n)$  and  $p(r_0), \dots, p(r_m)$ .
5.  $\{p(l_i) \mid 0 \leq i \leq n\} \cup \{p(r_j) \mid 0 \leq j \leq m\} = Q$

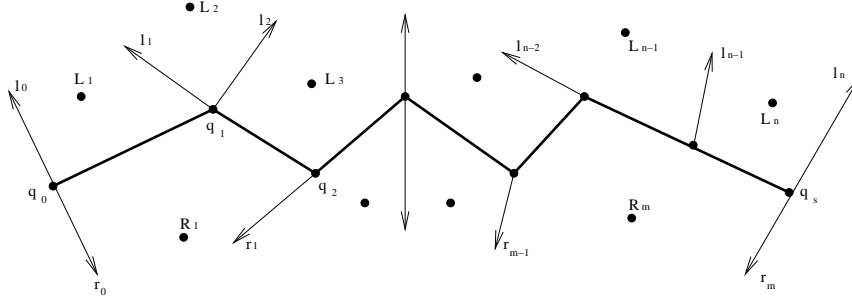


Figure 1: Example of a separation diagram. If the backbone is oriented from  $q_0$  to  $q_s$  then  $q_0$  is a negative wall relative to  $(l_0, r_0)$  and  $q_s$  is a positive wall relative to  $(l_n, r_m)$

6. For  $0 \leq i \leq s$ , if  $q_i$  is the endpoint of some ray  $t \in \Lambda \cup \Xi$ , then  $t$  does not contain any segment adjacent to  $q_i$  in  $C$ .
7.  $p(l_0) = p(r_0) = q_0$ .  $p(l_n) = p(r_m) = q_s$ .
8.  $l_0$  and  $r_0$  are collinear.  $l_n$  and  $r_m$  are collinear. If  $q_0 = q_s$  then  $\overline{l_n r_m} = \overline{l_0 r_0}$
9. In traversing  $C$  from  $q_0$  to  $q_s$ , one encounters, in order,  $l_0, l_1, \dots, l_n$  on the left and  $r_0, r_1, \dots, r_m$  on the right.
10.  $L = \{L_1, L_2, \dots, L_n\}$  and  $R = \{R_1, R_2, \dots, R_m\}$  are point sets such that each  $L_i$  ( $R_j$ ) lies in the interior of a finite or infinite convex region  $F(L_i)$  ( $F(R_j)$ ) which is the intersection of the halfplanes bounded by lines containing  $l_{i-1}$  ( $r_{j-1}$ ),  $l_i$  ( $r_j$ ), and each edge of  $C$  between  $p(l_{i-1})$  and  $p(l_i)$  ( $p(r_{j-1})$  and  $p(r_j)$ ).
11. Let  $[q_{k-1}, q_k]$  be one of the segments between  $p(l_{i-1})$  and  $p(l_i)$  ( $p(r_{j-1})$  and  $p(r_j)$ ). Then  $C$  does not intersect the interior of the triangle formed by  $L_i$ ,  $q_{k-1}$  and  $q_k$  (triangle formed by  $R_j$ ,  $q_{k-1}$  and  $q_k$ ).
12. A set  $E \subseteq \Lambda \cup \Xi$  of “excluded” rays whose endpoints are all distinct.
13. If  $e$  is an excluded ray,  $p(e) = q_j$ , and its complementary ray contains a consecutive chain of segments of  $C$  between  $q_j$  and some  $q_k$ , then all those segments are *excluded segments*.

**Lemma 4.2** For any separation diagram  $\Sigma$ , there exist  $x$  and  $y$  such that: (a)  $[L_x, R_y]$  intersects  $C$  in exactly one point; (b) that point is not on an excluded segment; (c)  $[L_x, R_y]$  does not intersect  $l_{x-1}$ ,  $l_x$ ,  $r_{y-1}$ , or  $r_y$ .

**Proof:** The proof is by induction on the size  $n+m$  of the separation diagram. The minimum size of a separation diagram is 2 with  $n = m = 1$ . In this case,  $C = [q_0, q_1]$  which by construction cannot be an excluded segment. The segment  $[L_1, R_1]$  obviously cannot intersect  $l_0$ ,  $l_1$ ,  $r_0$ , or  $r_1$  and intersects  $C = [q_0, q_1]$  in just one point.

Assume now that the lemma is true for all smaller sizes than the size of a given separation diagram. Assume w.l.o.g. that  $p(r_{m-1})$  is not closer on  $C$  to  $q_s$  than  $p(l_{n-1})$  is. This implies  $p(l_{n-1}) = q_{s-1}$ . Consider the following cases:

- Suppose  $[q_{s-1}, q_s]$  is an excluded segment. Suppose w.l.o.g. the associated excluded ray is  $l_i$ , with  $p(l_i) = q_k$ . If  $p(l_i) = q_0$  and  $i = 0$  then  $r_0$  would violate condition 6. Let  $j$  be the maximum value such that  $p(r_j)$  lies between  $q_0$  and  $q_k$ , inclusively, on  $C$ . Note that  $0 < i < n$  and  $0 \leq j < m$ . Form a new separation diagram: (1) delete  $q_{k+1}, \dots, q_s$ ; (2) delete  $l_{i+1}, \dots, l_n$  and  $r_{j+1}, \dots, r_m$ ; (3) delete  $L_{i+1}, \dots, L_n$  and  $R_{j+2}, \dots, R_m$ ; (4) add a new  $r_{j+1}$  which is the mirror of  $l_i$ . The region  $F(R_{j+1})$  in this new system is a proper superset of the original  $F(R_{j+1})$  produced by extending the boundary segment  $[q_k, q_s]$ , and can thus be seen to be a convex region containing  $R_{j+1}$ . The size of this new system is  $i + j + 1 < n + m$ , which by the induction hypothesis contains a suitable segment  $[L_x, R_y]$ . This segment is valid for the original system.
- Suppose  $[L_n, R_m]$  intersects  $l_{n-1}$  or  $r_{m-1}$ . The union  $U$  of the original  $F(L_n)$  and  $F(R_m)$  has exactly one reflex vertex at  $p(l_{n-1})$ ; since  $[L_n, R_m]$  leaves the interior of  $U$ , this means that it crosses  $l_{n-1}$  and that  $L_n$  and  $R_m$  are on opposite sides of the line that contains  $l_{n-1}$ . Therefore,  $l_{n-1} \neq l_0$ , and  $n > 0$ . Form a new separation diagram: (1) delete  $q_s, l_n$ , and  $L_n$ ; and (2) replace  $r_m$  by the mirror image of  $l_{n-1}$ . The region  $F(R_m)$  in this new system is a convex subset of the original  $F(R_m)$  produced by cutting along the line determined by  $l_{n-1}$ . The size of this new separation diagram is  $n - 1 + m$ , which by the induction hypothesis contains a suitable segment  $[L_x, R_y]$ . This segment is valid for the original system.
- Otherwise,  $[L_n, R_m]$  does not intersect  $l_{n-1}$  or  $r_{m-1}$ , then  $[L_n, R_m]$  intersects  $(q_{s-1}, q_s)$ , which is not an excluded segment. □

**Definition 4.3** *Given a planar overlay map  $(N, M, T)$ , an oriented separating path is a directed path in which each edge is obtained by orienting some subsegment of some barrier, and such that no connector of  $T$  intersects the path.*

**Definition 4.4** *Given an oriented separating path, consider two noncollinear consecutive edges  $(u, v)$  and  $(v, w)$  (see Figure 1).*

- *There are two faces adjacent to  $(u, v)$  which are also adjacent to  $v$ . If the sites in those two faces as well as  $u$  all lie within the same closed half-plane determined by the line through  $(v, w)$ , then  $v$  is called a positive wall.*
- *If  $w$  and the two sites in faces adjacent to  $(v, w)$  and  $v$  all lie within the same closed half-plane determined by the line through  $(u, v)$ , then  $v$  is called a negative wall.*

**Lemma 4.5** *No oriented separating path leads from a negative wall to a positive wall.*

**Proof:** Suppose there is such a path. Construct a separation diagram as follows: The backbone is the given path. The rays  $l_0$  and  $r_0$  are obtained by taking the line determining the halfplane at the negative wall and splitting it at the wall. However, if one of the adjacent sites at the  $w$  lies on the line itself rather than interior to the half-plane, rotate the rays slightly so that both sites lie within the interior of the half-plane determined by the rays. (By construction, it is impossible for both sites to lie on the line.) The rays  $l_n$  and  $r_m$  are constructed in a similar fashion from the positive wall. The sites of the separation diagram belong to faces adjacent to the backbone. The remaining rays are determined by taking each

barrier with a subsegment whose endpoint is a vertex of the backbone and whose interior does not intersect the backbone and extending those subsegments into rays. The excluded rays are those formed from barriers that intersect connectors.

To verify that this is a separation diagram, we must check the requirements listed in Definition 4.1. Most of the requirements correspond directly to features of the construction. That the faces are convex and adjacent to the backbone satisfies requirements 10 and 11. That the intersection of two barriers must involve an endpoint of one or both of them satisfies requirement 12.

Now Lemma 4.2 implies the existence of a connector crossing the given path, which contradicts the maximality of  $T$ .  $\square$

## 5 The structure of the planar overlay map

The following Lemma highlights an important feature of a planar overlay map and the graph created by the greedy algorithm. It proves that sections can only be nested one inside the other and that they admit a specific partial ordering. The proof uses a special construction called a *frame* that is actually a path separating two adjacent sections.

**Definition 5.1** *A frame is a set of segments of barriers that define a simple polygon on the planar overlay map, whose boundary does not intersect any connector (i.e. no section is located both on the inner side and outer side of the polygon/frame). The initial frame  $F_0$  is the boundary of the rectangle  $R$ , that contains all sites and all sections.*

**Lemma 5.2** *Exactly one section of a planar overlay map  $(M, N, T)$  touches each frame from the inside.*

**Proof:** Assume that more than one section touches some frame  $F$  from the inside. Take two adjacent segments on  $F$  that belong to two different sections  $S_i$  and  $S_j$ , where  $S_i$  has an edge on  $F$ . Call the intersection point of the segments (which is an intersection point of the two sections) the *start point*. Look at the boundary between the two sections that begins at their intersection point with  $F$ . This boundary is constructed from segments. Trace the boundary from the starting point and around one of the sections (assume it is around  $S_i$ ), until it hits  $F$  again. Call this intersection point the *end point*. By construction, the outer boundary of  $S_i$  is a simple closed polygonal curve.

By Lemma 4.5, it is not possible to orient the boundary between the two sections so that the start and end points form a positive and a negative wall. Therefore, at either the start or the end point,  $F$  has a reflex vertex. But the sites in the adjacent faces are located so that the vertex can serve as a negative or positive wall in either orientation of  $F$ . Therefore, orient the frame in either direction and observe the presence of a negative wall beginning at that point and a positive wall ending at that point. This contradicts Lemma 4.5.  $\square$

The above construction explains the relationship between sections and frames: each section is a polygon with holes, where the holes are filled by other sections and each frame is the outer boundary of exactly one section.



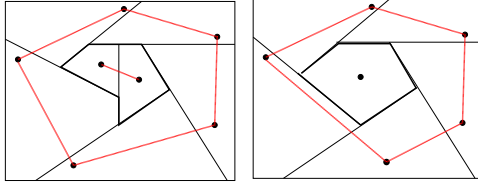


Figure 2: (a) A non convex frame. (b) A convex frame. The barriers are painted in black, the connectors in dotted thick red and the frame (the boundary of the section) in a bold black line. The connectors define the boundary of the buffer zone. A different set of choices in the greedy algorithm will result in only one section for (b) but will not change the structure of (a).

## 6 Connecting Sections

To build  $T$ , we need to connect sections until they are all connected to each other. We limit the set of connectors we use to connect two adjacent sections to those inside a specific region around the boundary between the two sections, called the *buffer zone*. Lemma 6.5 proves that such a connector can always be found. Lemma 6.6 proves that a barrier can participate in at most two buffer zones and hence the number of times it can be crossed is limited.

**Lemma 6.1** *No barrier intersects the interiors of two or more sections.*

**Proof:** This result follows from Lemma 4.5 (details omitted for brevity).  $\square$

**Lemma 6.2** *If  $P$  is a planar path from vertex  $s$  on one frame  $F_i$  to vertex  $t$  on another frame  $F_j$ , consisting of subsegments of barriers, then some connector intersects  $P$ .*

**Proof:** This result follows from Lemma 4.5 (details omitted for brevity).  $\square$

**Lemma 6.3** *For any frame  $F_i \neq F_0$ , there exists a closed curve of connectors enclosing  $F_i$  such that any other frame enclosed by this curve is nested inside  $F_i$ .*

**Proof:** Every frame  $F_i$ ,  $i \neq 0$ , is nested in another frame. By Lemma 6.2, every path from  $F_i$  to its enclosing frame is crossed by a connector. Consequently a closed curve of connectors encloses  $F_i$ . Consider the region formed by taking the intersection of the interiors of all closed curves of connectors enclosing  $F_i$  and deleting  $F_i$  and the region it encloses. If this region touches another frame  $F_j$ , then there is a path of subsegments of barriers connecting  $F_i$  to  $F_j$  such that no connector intersects this path. This would contradict Lemma 6.2.  $\square$

In order to connect any unconnected section  $S_i$  to its enclosing section  $S_j$  we need to build a connector (*bridge*) between a site on  $S_i$  to a site on  $S_j$ . For a *bridge* to cross as few barriers as possible, it must connect a site in  $S_j$  to a site on a zone that is ‘close’ to  $S_i$ :

**Definition 6.4** *The outer buffer zone  $B_{i_{outer}}$  of  $S_i$  is the intersection of all regions bounded by closed curves of connectors enclosing  $S_i$  with the region outside of  $S_i$ . The inner buffer zone  $B_{i_{inner}}$  of  $S_i$  is the intersection of  $S_i$  with the exteriors of all closed curves of connectors contained in  $S_i$ . The buffer zone  $B_i$  of  $S_i$  is the union of the inner and outer buffer zones.*

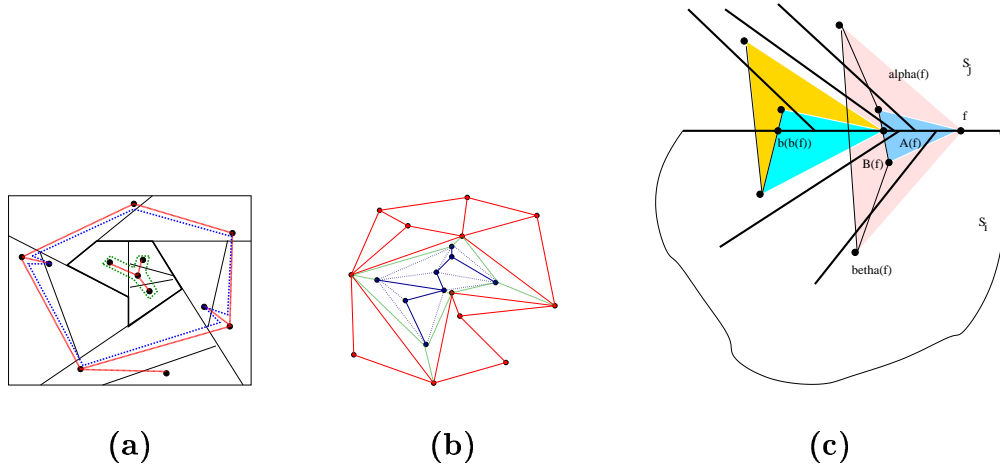


Figure 3: (a) The inner and outer polygons constructed in the proof of Lemma 6.5. (b) The red-blue triangulation in the proof of Lemma 6.5 with original connectors drawn in bold. Blue-red are thick dotted lines. (c) Construction of  $\alpha(f)$ ,  $\beta(f)$ ,  $A(f)$ ,  $B(f)$  as described in Section 8 (the two right triangles and their associated primary edges).  $\alpha(b(f))$ ,  $A(b(f))$  and  $b(b(f))$  are the two left triangles and the depicted point.

**Lemma 6.5** *Each unconnected section  $S_i$  of a planar overlay map  $(N, M, T)$ , must contain a site  $s_i$  that forms a bridge  $\beta$  with site  $s_o$  outside  $S_i$  such that  $\beta \subseteq B_i$ , where  $B_i$  is the buffer zone of  $S_i$ .*

**Proof:** Let  $P$  be the simple polygon defined by the set of connectors that are the boundary of the outer buffer zone of  $S_i$ . Take all the connectors that have a path to this set from the inside (*spokes*). Create a polygon  $P_{outer}$  by traversing  $P$  and  $P$ 's spokes. Let  $V_{outer}$  be the set of all sites on the boundary of  $P_{outer}$  and color them red. Consider the polygon  $P_{inner}$  created from the boundary of the inner buffer zone and its outer spokes in the same way. Let  $V_{inner}$  be the set of vertices on  $P_{inner}$  and color them blue.  $P_{outer}$  and  $P_{inner}$  define a polygon with a hole  $\hat{P}$  such that all vertices of the polygon are sites: vertices on the inner hole are (blue) sites from  $S_i$  and vertices on the outer boundary are (red) sites from the section enclosing  $S_i$  (see Figure 3 (a)). By construction there are no connectors inside  $\hat{P}$ .

Take the relative convex-hull  $Q_{inner}$  of  $P_{inner}$  relative to  $P_{outer}$ . Add as many disjoint blue-blue edges as possible, producing a triangulated blue polygon  $Q'_{inner} \subseteq Q_{inner}$ . Add as many disjoint red-red edges as possible outside  $Q_{inner}$ . Arbitrarily triangulate the rest by adding red-blue edges (Figure 3(b)). Every red-blue edge constitutes a *bridge*.  $\square$

**Lemma 6.6** *A barrier can participate in at most 2 buffer zones*

**Proof:** Assume some barrier  $b$  participates in more than two buffer zones. Then  $b$  must cross the boundary of one of the buffer zones twice, contradicting the principle of the greedy algorithm that no barrier be crossed more than once.  $\square$

## 7 The Crossing Number

**Theorem 7.1** *Given a set  $N$  of  $n$  barriers that partition the plane into  $m$  convex cells and a set  $M$  of  $m$  sites, exactly one in each cell, there exists a graph  $T^+$  spanning  $M$  that does*

not cross any barrier of  $N$  more than 3 times.

**Proof:** We construct a  $T^+$  as follows: *Step 1* is the greedy algorithm defined in Section 3. By the end of this step,  $T^+$  is a maximal set of planar connectors so that every barrier of  $N$  is crossed at most once and every connector crosses only one barrier. Any two adjacent sections induced by  $T^+$ , are nested one within the other (Lemma 5.2), creating a partial ordering of the sections.

*Step 2:* Continue as long as there is more than one section: pick an unconnected section  $S_i$ , add a bridge  $\beta$  to  $T^+$  that connects a pair of sites  $p_i$  (inside  $S_i$ )  $p_j$  (outside  $S_i$ ) that crosses only barriers that are in that buffer zone (Lemma 6.5).

After step 2, exactly one section remains, and all sites are connected to each other on the planar graph  $T^+$ . The additional crossings occur only on barriers that are part of a buffer zone. A barrier can participate in at most two buffer zones (Lemma 6.6). Hence a barrier can be crossed at most 3 times: once from step 1 and at most twice by step 2.  $\square$

## 8 Total cost

Consider a frame separating sections  $S_i$  and  $S_j$ . Let  $f$  be any point on the frame. We choose an associated potential bridge  $B(f)$  as follows. Let  $\alpha(f)$  be the closed triangle whose vertices are  $f$  and the two sites in the two faces adjacent to  $f$ . Let  $\beta(f)$  be the edge of this triangle which is opposite  $f$ . Call  $\beta(f)$  the *main* edge of  $\alpha(f)$  and the other two edges the *secondary* edges. If the interior of  $\alpha(f)$  contains no sites, let  $B(f) = \beta(f)$ . Otherwise, consider the convex hull of the sites in  $\alpha(f)$ , including the endpoints of  $\beta(f)$ . Traversing the boundary of this convex hull in the interior of  $\alpha(f)$  from one endpoint of  $\beta(f)$  to the other, one encounters sites from both  $S_i$  and  $S_j$ ; choose as  $B(f)$  any segment on the convex hull which connects a site in  $S_i$  to a site in  $S_j$ .

Define  $A(f)$  to be the triangle whose vertices are  $f$  and the intersections of the line determined by  $B(f)$  with the edges of  $\alpha(f)$ . The edge of this triangle which is an extension of  $B(f)$  will be known as its *main* edge and the other two its *secondary* edges (Figure 3 (c))

Note that the use of the convex hull ensures that the triangle  $A(f)$  is contained within the triangle  $\alpha(f)$  and that the secondary edges of  $A(f)$  are subsegments of the secondary edges of  $\alpha(f)$ . The secondary edges impose strong constraints exploited in the proofs below.

**Lemma 8.1** (a) *No barrier crosses a secondary edge except at the edge's endpoint at a frame.* (b) *Secondary edges do not intersect except at their endpoints.*

**Proof:** (a) By construction, the interior of each secondary edge lies entirely within one (convex) face, due to the convexity of the faces. Consequently, (b) an intersection of a secondary edge with the interior of another means two different sites (endpoints of those edges) within a single face.  $\square$

**Lemma 8.2** *No site lies in the interior of  $A(f)$ .*

**Proof:** Follows from  $B(f)$  lying on the boundary of the convex hull of the sites in  $\alpha(f)$ .  $\square$

For each point  $f$  on the frame, choose a point  $b(f)$  in the intersection of the frame with  $B(f)$ . Since the endpoints of  $B(f)$  are in different sections, there must be at least one such point. If there are multiple such points, choose  $b(f)$  arbitrarily. This process is applied iteratively. For example the notation  $b^3(f)$  denotes  $b(b(b(f)))$ .

**Lemma 8.3** *If  $b(f) \neq f$ , the interior of  $\alpha(b(f))$  shares no point with the interior of  $A(f)$ .*

**Proof:** The line determined by  $B(f)$  creates two half planes. Neither of the endpoints of  $\beta(b(f))$  lies in the same open half-plane as  $f$ . Suppose one does. By Lemma 8.2, it lies outside  $A(f)$ . Then the secondary edge  $[b(f), s]$  crosses one of the secondary edges of  $\alpha(f)$ , violating Lemma 8.1. Now, none of the vertices of  $\alpha(b(f))$  lies in the open half-plane that includes  $f$ , whereas by construction the interior of  $A(f)$  lies entirely within that half-plane. The result follows.  $\square$

**Lemma 8.4** *If  $b^3(f) \neq b^2(f) \neq b(f)$ , then  $A(b^2(f)) \cap A(f) = \emptyset$ .*

**Proof:** By Lemma 8.3,  $b^2(f)$  lies on the opposite side of the line determined by  $B(f)$  from  $f$ . If a secondary edge of  $A(b^2(f))$  crosses a secondary edge of  $A(f)$ , then Lemma 8.1 is violated. If a secondary edge of  $A(b^2(f))$  crosses  $B(f)$  and terminates within  $A(f)$ , then either Lemma 8.2 is violated or the associated secondary edge of  $\alpha(b^2(f))$  crosses a secondary edge of  $A(f)$ , violating Lemma 8.1. If  $B(b^2(f))$  is the only edge of  $A(b^2(f))$  which intersects  $A(f)$ , then a secondary edge of  $A(f)$  crosses  $B(b^2(f))$  and once again either Lemma 8.2 is violated or the associated secondary edge of  $\alpha(f)$  crosses a secondary edge of  $A(b^2(f))$ , violating Lemma 8.1. There are no other ways for the triangles to intersect.  $\square$

**Lemma 8.5** *If a barrier intersects the boundary of  $A(f)$  in two points, then one of them is  $f$ .*

**Proof:** If not, then one of the points is in a secondary edge of  $A(f)$ , violating Lemma 8.1.  $\square$

**Lemma 8.6** *Suppose there is no point  $f$  such that  $b(f) = f$ . Then there is a sequence of points on the frame  $(f_0, f_1, \dots, f_d = f_0)$ ,  $d \geq 3$ , such that  $f_i = b(f_{i-1})$  for all  $1 \leq i \leq d$ .*

**Proof:** Since there are only finitely many sites, the existence of such a sequence for some  $d$  is assured. By hypothesis,  $d \geq 2$ . But  $b^2(f) = f$  would mean the entire segment  $[f, b(f)]$  would lie within both  $A(f)$  and  $A(b(f))$ , contradicting Lemma 8.3.  $\square$

**Lemma 8.7** *At most  $\frac{5}{3}n$  total crossings suffice to construct a spanning tree of  $M$ .*

**Proof:** Construct  $T$ . Suppose there are  $K$  frames. Since each frame is enclosed by a cycle of connectors, we may delete  $K$  edges of  $T$  to produce a tree  $T^-$ .

For each frame, add a bridge determined as follows. If the frame contains a point  $f$  for which  $b(f) = f$ , use  $B(f)$ . Otherwise, consider  $B(f_0)$ ,  $B(f_1)$ , and  $B(f_2)$  from Lemma 8.6. Choose the bridge that intersects the fewest barriers, say it is  $B(f_i)$ . Note that by Lemmas 8.3 and 8.4, the interiors of no two of  $A(f_0)$ ,  $A(f_1)$ , and  $A(f_2)$  have any point in common.

We now classify the endpoints of all barriers to facilitate counting. If a barrier is crossed by a chosen bridge  $B(f_i)$  and its endpoint lies within  $A(f_i)$ , then we classify the endpoint as *crossed*. We classify all remaining endpoints of barriers as *uncrossed*. By our choice of bridges, and since the interiors of the triangles  $A(f_i)$  are distinct, at most  $\frac{1}{3}$  of the endpoints will be classified as crossed.

For each chosen bridge, each barrier crossed corresponds to an associated crossed endpoint, with one possible exception. The exception is the one barrier allowed under Lemma

8.5 which does not have an endpoint in the associated triangle. Since there are  $K$  chosen bridges, there are  $K$  of these exceptions. There are at most  $2n$  endpoints of barriers and therefore at most  $2n/3 + K$  crossings of barriers by bridges. The tree  $T^-$  creates  $n - K$  crossings. The total count is thus  $5n/3$ .  $\square$

**Lemma 8.8**  $\frac{5}{3}n$  total crossings are needed to construct a spanning tree of  $M$ .

**Proof:** Construct a honey-comb with  $n$  barriers as depicted in Figure 4 (a-b). Suppose  $T$  is a spanning tree of connectors. There are two types of sites: *blue* sites within the hexagonal faces and *red* sites within the triangular faces. The perimeter’s effect vanishes (like  $\frac{1}{\sqrt{n}}$ ) as the example grows in size. Ignoring the perimeter’s effect, the number of *red* sites is  $\frac{2}{3}n$  and the number of *blue* sites is  $\frac{1}{3}n$ . Say a connector is *type I* if it connects two blue sites, *type II* if it connects a blue and a red, and *type III* if it connects two red sites. Each type I connector crosses some barrier, and each type II or type III connector crosses two or more barriers. Consider the graph (a forest) formed from just the red sites and type III connectors. Because  $T$  is a spanning tree, each of its components is connected to some blue site with a distinct type II connector. Thus the combined number of type II and type III connectors is not less than the number of red sites. Thus  $T$  has  $n - 1$  connectors all of which cross some barrier and at least  $\frac{2}{3}n$  of which cross two barriers. This yields a total of at least  $\frac{5}{3}n - 1$  crossings.  $\square$

## 9 An Algorithm that Achieves at Most 3 Cuts per Barrier

We present a simple algorithm that creates in linear time a spanning tree with a cost of at most 3 cuts per edge, (not guaranteeing the bound of  $\frac{5}{3}n$  total cuts). The algorithm builds the tree of connectors by traversing the dual graph of the subdivision.

We assume the planar subdivision is represented as a Doubly-Connected-Edge-List (DCEL) where every cell contains exactly one site. In addition we have a list of *original barriers* where every edge of the DCEL has a pointer to an original barrier. The dual graph of the subdivision is easily accessible when using a DCEL.

### The Algorithm:

1. Initialize an adjacency list for the sites, which will be the base for the graph structure.
2. Walk on the DCEL. For every cell  $c_i$  and for all adjacent cells  $\{c_j\}$ , if a connector between  $c_i$  and  $c_j$  would cross the edge between the two cells (i.e. the connector would not cross more than one barrier) and if the edge between the two cells is linked to a barrier that is not yet marked as crossed, then: (1) Add the edge to the graph by inserting it into the adjacency list for both  $c_i$  and  $c_j$ ; (2) Mark the linked barrier as *crossed*.
3. Walk on the forest graph using any linear time walk (e.g. DFS). Walk on every component twice: In the first step mark every site as visited. In the second step check each site, until you locate a site adjacent to an unmarked site. These two sites are from

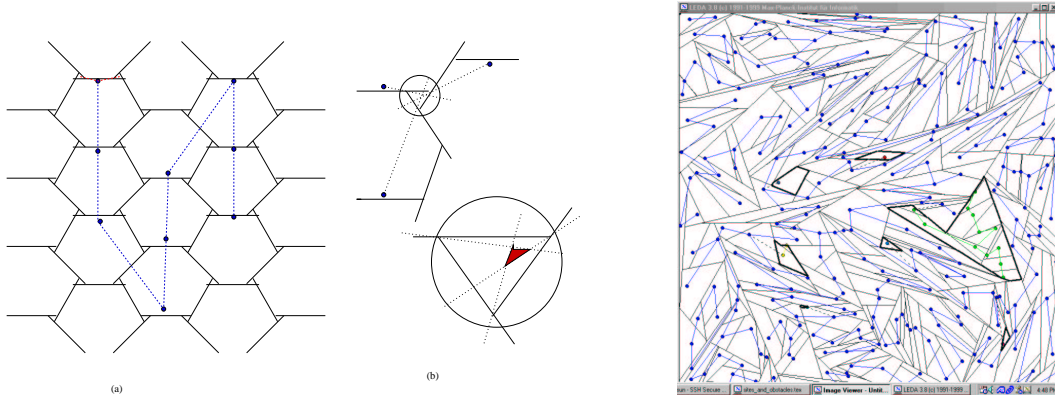


Figure 4: (a) Construction of the lower bound of two cuts per barrier and a total of  $\frac{5}{3}n$  using a *honey-comb*. (b) Enlargement of one section in the honey-comb. A connector to a site located in the shaded region will always cross at least two barriers. (c) An example of an output of the program described in Section 10 consisting of 11 sections, frames drawn in bold.

different sections and, since they share a barrier, can be safely connected. Connect the two sites by adding the edge to the forest. Repeat the graph walk for the tree rooted at the newly added site.

The algorithm visits all cells and all sites. In the first step it connects two sites only if their adjacent barrier is not already crossed. In the second step additional connectors are added but only across segments between sections, yielding at most 3 cuts per edge. The algorithm's time complexity is linear in the size of the subdivision,  $n + m$ .

## 10 Experimental Results

A 'brute force' computer program was built according to the asymptotically slower algorithm described in Section 7 and was tested on planar subdivisions created at random. The code was written in C++ using the LEDA graphics library for visual output. The program has quadratic time complexity, when the extended plane partition is given.

To create a problem instance, the program creates small segments at randomly chosen angles from the horizontal. Then it visits all segments repeatedly extending their endpoints bit by bit until every endpoint terminates in another segment.

We created 100,000 random problem instances of 200 barriers. We found that the planar subdivisions generally had one large outer section and only a few small other sections which were easily bridged. The largest number of sections we found was 17. On average, in all instances generated, each barrier was crossed around 1.003 times! In the worst case, there were 214 crossings of the 200 barriers, an average of 1.07 per barrier. For one unusually rich randomly generated example, see Figure 4 (c).

## 11 Summary and future work

We defined a restricted version of a problem posed by Snoeyink. The original problem has known lower bounds of 2 cuts per edge and  $2n - 2$  total cost; upper bounds of 4 and  $4n$ , respectively. We proved lower bounds of 2 and  $\frac{5}{3}n - o(1)$ , and upper bounds of 3 and  $\frac{5}{3}n$ . Our experience persuades us that Theorem 7.1 can be improved:

**Conjecture 11.1** *Given a set  $N$  of  $n$  barriers that partition the plane into  $m$  convex cells and a set  $M$  of  $m$  sites, exactly one in each cell, there exists a graph  $T^+$  spanning  $M$  that does not cross any barrier of  $N$  more than 2 times.*

The original problem remains open and the techniques described here do not apply directly because of the existence of empty faces (holes). We are working to apply generalized versions of the *planar overlay map* and the *separation diagram* to the case of a planar graph with holes, as a first step towards solving the problem of a planar subdivision. Furthermore, we are also studying the same problems in the *easier* case, where the sites' locations are not set in advance and can be placed by the algorithm anywhere in a cell. Another open problem is devising an optimal algorithm to construct a spanning tree of total low cost.

**Acknowledgement** The authors would like to thank Jack Snoeyink for his detailed reading and insightful comments.

## References

- [1] T. Asano, M. de Berg, O. Cheong, L.J. Guibas, J. Snoeyink, and H. Tamaki. Spanning trees crossing few barriers. In *Proceedings of the 15th Ann. Symp. on Comp. Geom.*, pages 41–48, New York, 1999.
- [2] M. de Berg, M. de Groot, and M. Overmars. New results on binary space partitions in the plane. *Comput. Geom.*, 8(6):317–333, 1997.
- [3] M. de Berg, M. de Groot, and M. Overmars. Perfect binary space partitions. *Comput. Geom.*, 7(1-2):81–91, 1997.
- [4] C. Duncan. Multi-way space partitioning trees. In *Proc. 12th Annual Fall Workshop on Computational Geometry*, DIMACS, NJ, 2002.
- [5] H. Edelsbrunner, L. Guibas, and J. Stolfi. Optimal point location in a monotone subdivision. *SIAM J. Comput.*, 15(2):317–340, 1983.
- [6] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. *ACM Trans. on Graphics*, 4(2):74–123, 1985.
- [7] M. Hoffmann and C.D. Toth. Connecting points in the presence of obstacles in the plane. In *Proceedings of the 14th Canadian Conference on Computational Geometry*, pages 63–67, 2002.
- [8] D.G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM J. Comput.*, 12(1):28–35, 1983.

- [9] M. Paterson and F. Yao. Efficient binary space partitions for hidden-surface removal and solid modeling. *Discrete Comput. Geom.*, 5(5):485–503, 1990.
- [10] N. Sarnak and R.E. Tarjan. Planar point location using persistent search trees. *Commun. of the ACM*, 29(7):669–679, 1986.
- [11] J. Snoeyink. Open problems session, August 1997. 9th Canadian Conf. on Comp. Geo.
- [12] J. Snoeyink. Private communications, 1997.
- [13] J. Snoeyink and M. van Kreveld. Linear-time reconstruction of Delaunay triangulations with applications. In *Algorithms—ESA '97 (Graz)*, volume 1284 of *Lecture Notes in Comput. Sci.*, pages 459–471. Springer, Berlin, 1997.
- [14] C. Toth. A note on binary plane partition. In *Proc. 17th ACM Symp. Comp. Geom.*, pages 151–156, Medford, MA, 2001.
- [15] C. Toth. Binary space partitions for line segments with a limited number of directions. In *Proc. 13th SIAM-ACM Symp. on Disc. Alg. (SODA)*, pages 465–471, San-Francisco, CA, 2002.