Growing a Tree from its Branches*

Prosenjit Bose and Godfried Toussaint

School of Computer Science
McGill University
3480 University Street
Montréal, Québec, Canada
H3A 2A7

Abstract

Given a set of n disjoint line segments in the plane, we show that it is always possible to form a spanning tree of the endpoints of the segments, such that each line segment is an edge of the tree and the tree has no crossing edges. We also show that the maximum degree of a node in the minimum weight spanning tree formed by these line segments is seven, and that there exists a set of line segments achieving this bound. The time complexity of finding the minimum weight spanning tree is $O(n \log n)$. Finally, we show an $\Omega(n \log n)$ lower bound for this problem, establishing the optimality of the algorithm.

1 Introduction

Given a set of disjoint line segments, determining whether the set admits certain combinatorial structures has received considerable attention. ElGindy and Toussaint [3] have shown that every set of line segments can be triangulated. Another structure which has been studied is the simple circuit or polygon through a set of line segments. Rappaport [9] has shown that deciding whether a set of line segments admits a simple circuit is an NP-complete problem. Certain special cases have also been studied. Avis and Rappaport [1] give an $O(n^4)$ time and $O(n^2)$ space algorithm to decide whether a set of line segments admits a simple monotone circuit. Rappaport, Imai, and Toussaint [10] have shown that the decision problem is $O(n \log n)$ when every line segment in the set has at least one endpoint on their convex hull (such a configuration is known as a *convexly independent* set of line segments). Although not every convexly independent set of line segments admits a simple circuit, Mirzaian [7] has shown that such a set always admits a simple polygon such that the line segments are either part of the boundary of the polygon or form internal

*Research supported in part by an NSERC PGS2 scholarship, NSERC-A9293 and FCAR-EQ1678

diagonals. Mirzaian's result does not hold for arbitrary sets of disjoint line segments, but it is conjectured that the result is true if the line segments are afforded the additional freedom of forming external diagonals of the polygon. O'Rourke and Rippel [4] proved the hamiltonicity of the visibility graph of certain restricted classes of line segments.

Surprisingly, the problem of determining whether a set of line segments admits a spanning tree on the endpoints such that each line segment is an edge of the tree and the tree has no crossing edges has not been studied at all. This notion leads to several intriguing questions. Does a set of line segments always admit such a tree? Are there efficient algorithms for finding such a tree if one exists? Is there a bound on the maximum degree of such a tree? We answer all these questions in the affirmative.

2 Notation and Definitions

Let us first introduce some terminology. Most of the geometric and graph theoretic terminology used is standard and for details, we refer the reader to O'Rourke [5], Bondy and Murty [2], and Preparata and Shamos [6]. We begin by reviewing some of the main geometric and graph theoretic terminology used in this paper.

Let E^2 denote the Euclidean space of dimension two. A domain D in E^2 is *convex* if, for any two points q_1 and q_2 in D, the segment $\overline{q_1q_2}$ is entirely contained in D. The *Convex Hull* of a set of points S in E^2 is the boundary of the minimum area convex domain in E^2 containing S.

Two line segments in the plane are *disjoint* if they do not intersect. A set of line segments is disjoint if every pair of line segments in the set is disjoint. Given a set, L, of disjoint line segments in the plane, we say that two points in the plane are *visible* with respect to L, if the open line segment determined by the two points does not intersect any line segment in L. Throughout this paper, we assume that the endpoints of the line segments are in general position, i.e., no three are collinear and no four are co-circular.

A *graph* $G = (V(G), E(G))$ consists of a finite nonempty set $V(G)$ of $vertices$, and a set $E(G)$ of unordered pairs of vertices known as *edges*. A *vertex disjoint path* in a graph, G, is a finite non-null sequence $P = v_1e_1v_2e_2 \dots e_kv_{k+1}$ where the edges $e_1, e_2 \dots e_k$, and the vertices $v_1, v_2 \dots v_{k+1}$ are distinct. The vertices v_1 and v_{k+1} are known as the *origin* and the *terminus* of the path. A *cycle* is a path whose origin and terminus are the same. An *acyclic* graph is a graph that contains no cycles. A *connected* graph is a graph which has a path between every pair of vertices. A *tree* is a connected acyclic graph. A *subgraph* of a graph G, is a graph whose vertex set and edge set are subsets of the vertex and edge set of G. A *spanning* subgraph of a graph G, is a subgraph of G whose vertex set is the same as the vertex set of G. If a spanning subgraph of a graph G is a tree, then it is known as a spanning tree of G.

Given a set L of disjoint line segments, we define the *visibility* graph of L, denoted as $V(L)$, as follows. The endpoints of the line segments in L correspond exactly to the vertices of $V(L)$. If two endpoints are visible with respect L, then their corresponding vertices are adjacent in $V(L)$.

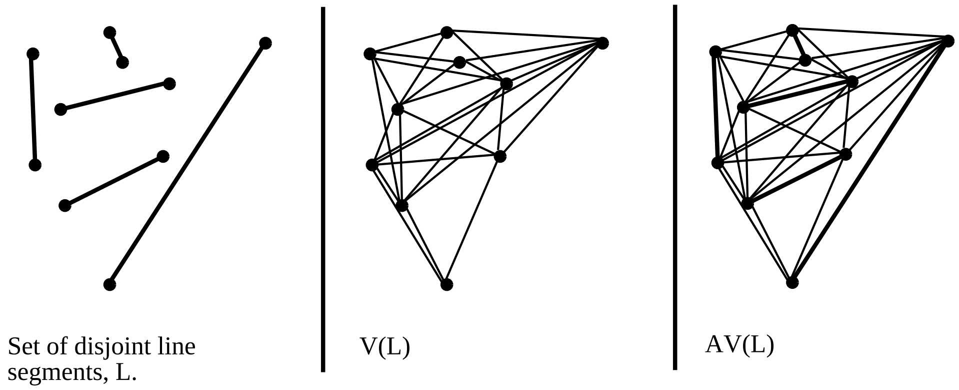

Figure 1:

The *augmented visibility* graph of the set L, denoted $AV(L)$, is the same as $V(L)$, with the addition that every line segment in L has a corresponding edge in $AV(L)$. Thus, $AV(L)$ contains two types of edges: *line segment edges*, and *visibility edges* (refer to figure 1).

3 Existence of Spanning Trees

We begin by showing that every set of disjoint line segments admits a spanning tree, such that every line segment is a tree edge and the tree has no crossing edges. By this, we mean that the augmented visibility graph of the set of line segments contains a spanning tree which has a planar embedding and includes every edge corresponding to a line segment in L. A spanning tree that has this property is said to *encompass* the set L.

First, we show that the augmented visibility graph of a set of disjoint line segments, L, is connected. Since a triangulation of L is a subgraphs of $AV(L)$ and a triangulation is connected, this result follows from [3]. However, we include an alternate direct proof here for completeness.

Lemma 1 *The augmented visibility graph of a set of disjoint line segments is connected.*

Proof: By induction on the number of line segments (refer to figure 2).

Basis: $n = 1$. When there is only one line segment, the graph is trivially connected.

Inductive Hypothesis: Given a set of $n \leq k$, $k \geq 1$, disjoint line segments, the augmented visibility graph is connected.

Inductive Step: $n = k + 1$. Pick a line segment s which has at least one endpoint e_1 on the convex hull. By removing the segment s from the set L, we have a smaller set L'. By the inductive hypothesis, $AV(L')$ is connected. By inserting s into the set L', the graph might become disconnected because some visibility edges may no longer exist. Consider an arbitrary visibility edge, $\overline{ab}$, in $AV(L')$ that is not in $AV(L)$. We will show that there exists a path from a to b in $AV(L)$. Therefore, $AV(L)$ must be connected, since there exists a path from the two endpoints of any visibility edge deleted from $AV(L')$ by the insertion of s, and s itself is connected by at least one endpoint on the convex hull.

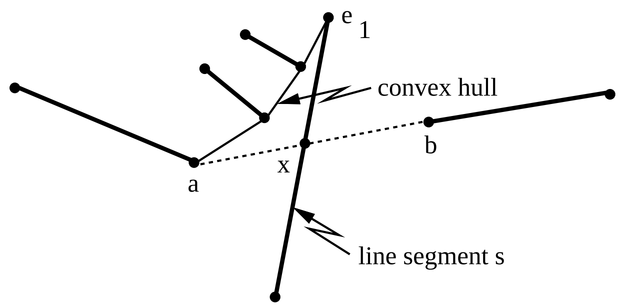

Figure 2:

The line segment $\overline{ab}$ must intersect s at some point x. Consider the convex hull of the endpoints inside the triangle(a, x, e_1), excluding x. Either the convex hull is a single edge from a to e_1 when the triangle is empty, or there is a path from a to e_1 when the triangle contains at least one endpoint. Similarly, either there is a single edge from b to e_1, or there is a path from b to e_1. ■

Lemma 2 *[2] Every connected graph contains a spanning tree.*

What remains to be shown is that this connected graph contains the appropriate spanning tree. Let us modify $AV(L)$, by giving weights to the edges in the graph. Each edge representing a line segment is assigned weight 0 and every other edge is assigned weight 1. In $AV(L)$, the set of edges representing the line segments do not form any cycles since the line segments are disjoint by definition. Therefore any minimum weight spanning tree of the modified $AV(L)$ will include all the line segment edges, and the existence of the desired spanning tree is guaranteed. We have therefore established the following result.

Lemma 3 *Every set of disjoint line segments admits an encompassing spanning tree.*

The spanning tree that is found in the augmented visibility graph, can be embedded directly on the set of line segments from which the graph arises. The question then becomes: can the spanning tree always be embedded without intersection? For example, the tree illustrated in figure 3 has an intersection, but an alternate tree can be found.

Lemma 4 *[3] The augmented visibility graph of a set of disjoint line segments has a near-triangulated planar subgraph containing all the line segment edges.*

A near-triangulated planar graph is a planar graph where every face is a triangle except possibly the outer face. In fact, in [3], it is shown how to triangulate a set of disjoint line segments in $O(n \log n)$ time. By assigning a weight 0 to each line segment edge, and

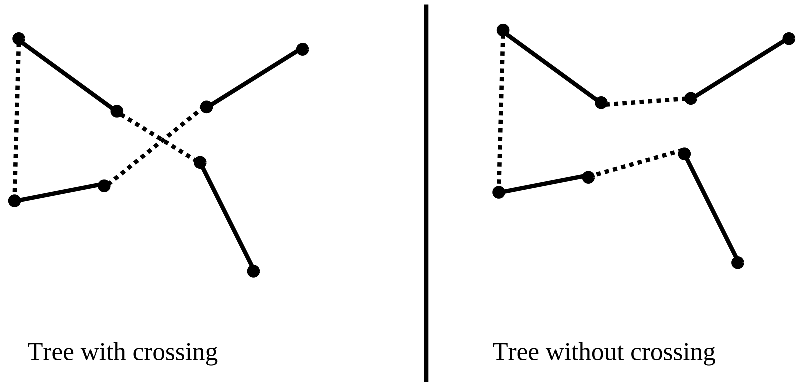

Figure 3:

a weight 1 to every other edge of the triangulation, we see that any minimum weight spanning tree will be an intersection-free encompassing spanning tree. Since there are a linear number of edges in a planar graph, the minimum weight spanning tree can be constructed in $o(n \log n)$ time using standard algorithms ([6], [13]). Therefore we have the following theorem.

Theorem 1 *Every set of disjoint line segments admits an encompassing spanning tree with no intersecting edges, and such a tree can be constructed from the set of line segments in $O(n \log n)$ time, where n is the number of line segments.*

4 Bounded Degree Spanning Trees

In this section, we investigate properties of the minimum weight spanning tree of a set of disjoint line segments. Let us first define the assignment of weights. In the augmented visibility graph of a set of disjoint line segments, a weight of zero is given to every edge corresponding to a line segment, and the euclidean distance between the endpoints represents the weight of every visibility edge. Therefore the minimum weight spanning tree of a set of disjoint line segments is an encompassing spanning tree.

Since we are considering the whole augmented visibility graph when determining the minimum weight spanning tree as opposed to an intersection-free subgraph (triangulation) as was done in the previous section, it is no longer clear whether the minimum weight spanning tree is intersection-free.

Lemma 5 *The minimum weight spanning tree of a set of disjoint line segments is intersection-free.*

The question now is whether we can bound the maximum degree of a node in the minimum spanning tree of a set of disjoint line segments. It is not unreasonable to believe that the maximum degree can be bounded since we know the following.

Lemma 6 *[8] Every finite set of points S in the plane admits a minimum spanning tree whose maximum node degree is five.*

In a finite set of points in general position, every point can see every other point, which implies a quadratic number of edges in the visibility graph of the set of points, but in a disjoint set of line segments, there are some points which do not see each other. In fact, it is shown in [11] that $5n - 4$ is a tight lower bound on the number of edges in the augmented visibility graph of a set of n disjoint line segments. Although, there may be only a linear number of edges in the augmented visibility graph, we establish the following lemma.

Lemma 7 *The minimum weight spanning tree of a set of disjoint line segments has maximum degree seven.*

Proof: Suppose that the minimum weight spanning tree, T, had a vertex, v, with degree $d(v) \geq 8$; then vertex v must be adjacent to one line segment edge and at least seven visibility edges. Thus, there must be two adjacent visibility edges such that the angle between them is less than $\frac{\Pi}{3}$. Let $\overline{va}$ and $\overline{vb}$ be two such edges and refer to figure 4. Without loss of generality, assume that $||\overline{vb}|| \geq ||\overline{va}||$. Let α be the angle between the two edges.

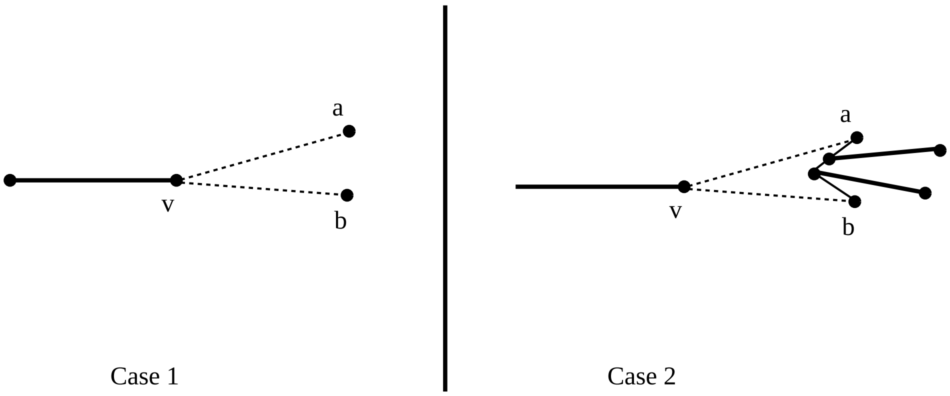

Figure 4:

Case 1: Suppose a sees b. Since $\alpha < \frac{\Pi}{3}$, the edge $\overline{vb} > \overline{ab}$. Thus, the tree $T' = T - \overline{vb} + \overline{ab}$ has lower total weight, contradicting the fact that T has minimum weight.

Case 2: Suppose a does not see b. There must be at least one line segment blocking visibility. That line segment must therefore intersect $\overline{ab}$. Furthremore, since $\overline{va}$ and $\overline{vb}$ are visibility edges, one endpoint of the line segment must lie in the triangle(a, v, b). Consider the convex hull of all the endpoints inside the triangle(a, b, v) including a and b. Aside from a and b, let the other convex hull vertices be denoted by $c_1, c_2, \ldots, c_k$, where k is two less than the total number of convex hull vertices. Since there is at least one endpoint in the triangle, $k \geq 1$. The convex hull consists of the edge $\overline{ab}$ and a path $\Pi = \overline{bc_1}, c_1, \overline{c_1c_2}, c_2, \ldots, \overline{c_ka}$.

Removing the edge $\overline{vb}$ from T, disconnects the tree into two components. One of the components contains b and the other contains a, since there is an edge between v and a. Therefore, one of the convex hull edges must connect the two components, since Π connects a to b. This edge, let us denote it by e_c, is smaller than $\overline{vb}$ since it is wholy contained in triangle(a, b, v), and $\overline{vb}$ is the triangle's longest side due to the fact that $\alpha < \frac{\pi}{3}$. Thus, the tree $T' = T - \overline{vb} + e_c$ has lower total weight, contradicting the fact that T is minimum weight.

■

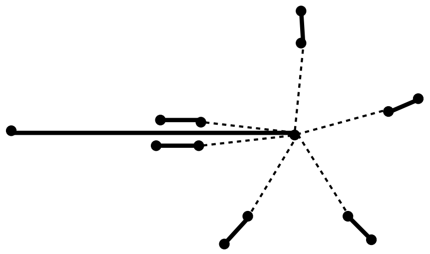

Figure 5:

The bound on the maximum degree in the minimum spanning tree is tight because the set of line segments in figure 5 exhibits a minimum encompassing spanning tree with maximum degree seven. The angle between every pair of visibility edges adjacent to the vertex of degree seven is greater than $\frac{\pi}{3}$ except for the pair whose visibility is blocked by the line segment. We have therefore proved the following theorem.

Theorem 2 *The minimum weight spanning tree of a set of line segments contains no intersecting edges and has maximum degree seven. There exists a set of line segments achieving the maximum degree.*

5 Algorithms

In this section, we present an $O(n \log n)$ time algorithm to find the minimum weight spanning tree of a set of disjoint line segments.

Let the euclidean distance between two points p and q be denoted by $d(p, q)$. The $lune(p, q)$ of two points p and q is the interior of the of the intersection of two circles with radii equal to $d(p, q)$ centered at p and q, respectively. Notice that for any point x lying inside $lune(p, q)$, we have that both $d(x, p) < d(p, q)$ and $d(x, q) < d(p, q)$.

We now define an object known as the *constrained relative neighborhood graph*. The relative neighborhood graph constrained to a set of line segments L, denoted as $CRNG(L)$, is a graph defined in the following way. A node of $CRNG(L)$ corresponds to an endpoint of a line segment in L. An edge exists between two nodes if and only if their corresponding endpoints e_1 and e_2 have one of the following properties:

2. No endpoint lying in $lune(e_1, e_2)$ is visible from both e_1 and e_2.

Lemma 8 *The minimum weight spanning tree of a set of disjoint line segments, L, is a subgraph of the constrained relative neighborhood graph on that set.*

Proof: Suppose an edge of the minimum weight spanning tree, T, is not an edge of the constrained relative neighborhood graph. This edge must be a visibility edge, since by definition all line segment edges are contained in the $CRNG(L)$.

Let e be such an edge, with endpoints a and b. Since e is not an edge in the $CRNG(L)$, the $lune(a, b)$ must contain an endpoint x which is visible to both a and b. The removal of e from the tree, disconnects the tree into two trees T_1 and T_2, with $a \in T_1$ and $b \in T_2$. Since x was in T, it must be in one of the two components. Without loss of generality, let $x \in T_1$. Thus, the tree $T' = T - e + \overline{xb}$ has lower weight than T contradicting the fact that T has minimum weight. ■

Su and Chang [12] presented an algorithm to compute the constrained relative neighborhood graph of a set of disjoint line segments in $O(n \log n)$ time. Since there are a linear number of edges in such a graph, we can then apply any $o(n \log n)$ time minimum spanning tree algorithm ([6], [13]) to the resulting constrained relative neighborhood graph. Therefore we have the following theorem.

Theorem 3 *The minimum weight spanning tree of a set of disjoint line segments can be computed in $O(n \log n)$ time.*

The problem of finding the minimum spanning tree of a set of disjoint line segments requires time $\Omega(n \log n)$.

Theorem 4 *The problem of sorting n real numbers is $O(n)$ transformable to the problem of finding a minimum spanning tree of a set of disjoint line segments; thus, finding the minimum spanning tree of a set of disjoint line segments requires $\Omega(n \log n)$ time.*

Thus, the algorithm for finding the minimum weight spanning tree of set of disjoint line segments is optimal.

6 Conclusion

In this paper, we have shown that a disjoint set of line segments always admits an encompassing spanning tree with no crossing edges. This tree can be computed in $O(n \log n)$ time. We have also shown that the minimum weight spanning tree of the set of line segments contains no crossing edges and has maximum node degree seven. The encompassing minimum spanning tree is a subgraph of the constrained relative neighborhood graph and can be computed in time $O(n \log n)$. Finally, we have shown a lower bound of $\Omega(n \log n)$ for the problem establishing that our algorithm is optimal.

7 Acknowledgments

The authors wish to thank Anna Lubiw, Eduardo Rivera-Campo, and Jorge Urrutia for interesting discussions on the topic, David Thompson for providing a stimulating research environment and one of the referees for pointing out an error in a preliminary version of this paper.

References

[1] D. Avis and D. Rappaport. Computing Monotone Simple Circuits in the Plane. *Computational Morphology*. Elsevier Science, G.T. Toussaint (editor), North Holland, 1988.

[2] J.A. Bondy and U.S.R. Murty. *Graph Theory with Applications*. Elsevier Science, New York, New York, 1976.

[3] H. ElGindy and G. Toussaint. Efficient Algorithms for Inserting and Deleting Edges from Triangulations. *Proc. International Conference on Foundations of Data Organization*, Kyoto, Japan, 1985.

[4] J. O'Rourke and J. Rippel. Two Segment Classes with Hamiltonian Visibility Graphs. Technical Report 15, Smith College, 1992.

[5] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, New York, 1987.

[6] F. Preparata and M. Shamos . *Computational Geometry: An Introduction*, Springer-Verlag, New York, New York, 1985.

[7] A. Mirzaian. Hamiltonian Triangulations and Circumscribing Polygons of Disjoint Line Segments. *Computational Geometry: Theory and Applications,* 1992. To appear. First presented in 1990 at the Canadian Computational Geometry Conference.

[8] C. Monma and S. Suri. Transitions in Geometric Minimum Spanning Trees. Bell Communications Research Technical Report, 1991.

[9] D. Rappaport. Computing simple circuits from a set of line segments is NP-complete. *the 3rd ACM Symposium on Computational Geometry*, Waterloo, Ontario, pp. 52-60, 1987.

[10] D. Rappaport, H. Imai, and G.T. Toussaint. Computing Simple Circuits from a Set of Line Segments. *Discrete and Computational Geometry*, **5, 3**, pp. 289-304, 1990.

[11] X. Shen and H. Edelsbrunner. A Tight Lower Bound on the Size of Visibility Graphs, *Information Processing Letters*, **26**, pp. 61-64, 1987.

[12] T. Su and R. Chang. Computing the constrained relative neighborhood graphs and constrained Gabriel graphs in Euclidean plane. *Pattern Recognition*, **24, 3**, pp. 221-230, 1991.

[13] A. Yao. An $O(E \log \log V)$ algorithm for finding minimum spanning trees, *Information Processing Letters*, **4**, pp. 21-23, 1975.