Disjoint Segments have Convex Partitions with 2-Edge Connected Dual Graphs

Nadia M. Benbernou* Erik D. Demaine† Martin L. Demaine‡ Michael Hoffmann§
Mashhood Ishaque¶ Diane L. Souvaine‖ Csaba D. Tóth**

Abstract

The empty space around n disjoint line segments in the plane can be partitioned into $n+1$ convex faces by extending the segments in some order. The *dual graph* of such a partition is the plane graph whose vertices correspond to the $n+1$ convex faces, and every segment endpoint corresponds to an edge between the two incident faces on opposite sides of the segment. We construct, for every set of n disjoint line segments in the plane, a convex partition whose dual graph is 2-edge connected.

1 Introduction

Noncrossing line segments are fundamental in computational geometry. Algorithms on disjoint line segments—in visibility, motion planning, graph drawing [5, 8, 9, 10, 11]—often consider a convex partition and its dual graph.

A set of n disjoint line segments in the plane and a permutation π of the $2n$ segment endpoints define a partition of the plane into convex faces: extend the segments beyond their endpoints one-by-one in the order given by π until they hit another segment, a previous extension, or infinity. If no three segment endpoints are collinear, then we obtain $n+1$ convex faces for any permutation π. In general, the convex partition depends on the order π in which the extensions are drawn. All permutations lead to the same convex partition if and only if no extension meets any other extension.

In the *dual graph* of a convex partition, the vertices correspond to the $n+1$ convex faces, and every segment endpoint corresponds to an edge between the two incident faces on opposite sides of the segment (see Fig. 1). Two vertices are connected by a *double edge* if and only if an entire segment (and both endpoints) lie on the common boundary of the corresponding convex faces.

*Massachusetts Institute of Technology, nbenbern@mit.edu
†Massachusetts Institute of Technology, edemaine@mit.edu
‡Massachusetts Institute of Technology, mdemaine@mit.edu
§ETH Zürich, hoffmann@inf.ethz.ch
¶Tufts University, mishaq01@cs.tufts.edu
‖Tufts University, dls@cs.tufts.edu
**Massachusetts Institute of Technology, toth@math.mit.edu

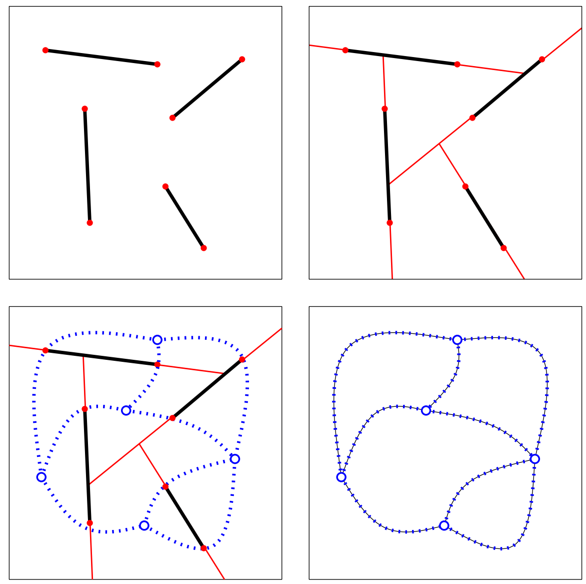

Figure 1: Disjoint line segments, a convex partition, and the corresponding dual graph.

Kano [6, 7] worked on various discrete geometry problems involving monochromatic partitions of geometric graphs. Motivated by questions on compatible plane matchings, he and others [3] posed the following open problem:

Problem 1 *Does every finite set of disjoint segments in the plane have a convex partition such that the dual graph can be decomposed into two spanning trees?*

In the special case that the convex partition is unique (that is, does not depend on the permutation π), the answer is *yes*: color every *left* segment endpoint red and every *right* segment endpoint blue—it is not difficult to see that both the red and the blue subgraphs are spanning trees [5]. This simple coloring scheme, however, does not work in general. There are sets of (already as few as four) segments where the subgraphs corresponding to either all the left or all the right segment endpoints is disconnected for any convex partition.

It is easy to construct *a* convex partition for n disjoint line segments in $O(n \log n)$ time with two sweep-line algorithms: first a left-to-right sweep extends all segments beyond their right endpoints, and then a right-to-left sweep extends all segments beyond their left endpoints. Whenever two right (left) extensions meet, one must have priority over the other: these choices give different variants of this partitioning scheme. There are sets of segments (Fig. 2) for which any variant of this simple scheme leads to a dual graph with a bridge, making a decomposition into two spanning trees impossible.

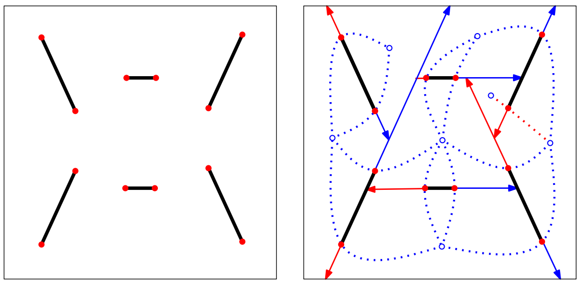

Figure 2: Six segments for which any variant of the sweepline scheme leads to a dual graph with a bridge.

In this note we show how to find a convex partition whose dual graph has no bridges:

Theorem 1 *For any finite set of disjoint line segments in the plane, one can construct a convex partition whose dual graph is 2-edge connected.*

We give an algorithm for constructing such a convex partition. Its runtime is $O(n^{4/3+\varepsilon})$ for n input segments, dominated by $2n$ ray shooting queries among $O(n)$ polygonal objects, one for each segment endpoint.

Theorem 1 does not solve Problem 1. Fig. 3 depicts four segments and a convex partition whose dual graph is 2-edge connected, yet cannot be decomposed into two spanning trees. This convex partition, however, *cannot* be produced by our algorithm. We conjecture that the dual graph of the convex partition we propose can always be decomposed into two spanning trees.

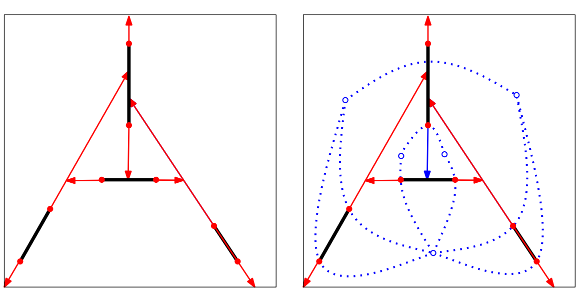

Figure 3: A convex partition whose dual graph is 2-edge connected but cannot be split into two spanning trees.

For n disjoint segments in the plane, it is natural to define a smaller family of convex partitions, where the two endpoints of each segment should be consecutive in the permutation π (practically, extending each segment in both directions at the same time). We do not know whether Theorem 1 remains true under this restriction.

Problem 2 *Is there a permutation of every finite set of disjoint segments such that, extending the segments in both directions in this order, the dual graph of the resulting convex partition is 2-edge connected?*

2 Partition by Induction

We prove Theorem 1 algorithmically. We first show that one can apply induction in a certain sense. It is enough to prove the following.

Theorem 2 *For any finite set S of disjoint line segments in the plane, one can find a nonempty subset $S' \subseteq S$ and construct a convex partition P' of S' such that the dual graph of P' is 2-edge connected and every segment of $S \setminus S'$ lies in the interior of a face of P'.*

Let us show that Theorem 1 follows from Theorem 2:

Proof of Theorem 1. Proceed by induction on S. Let S be a set of n disjoint line segments, and assume that any set of less than n segments has a convex partition whose dual graph is 2-connected. By Theorem 2 there is a convex partition P' of a nonempty subset S' of segments. If $S = S'$, then the proof is complete. Assume that $S \setminus S' \neq \emptyset$. By induction, the (possibly empty) subset of segments lying in each face $v \in P'$ has a convex partition (of that face) whose dual graph $G(v')$ is also 2-connected. The partitions of the faces of P' jointly give a convex partition P of S.

The dual graph G of P is obtained from the dual graph G' of P' by replacing each vertex $v' \in V(G')$ with a 2-edge connected graph $G(v')$ and replacing every edge $u'v' \in E(G')$ with an edge between some vertices of $G(u')$ and $G(v')$. We claim that any graph G obtained in this way is 2-edge connected.

Indeed, it is enough to show that there are two edge-disjoint path between any two vertices, u and v, of G. This is obvious if both u and v are vertices of a subgraph $G(w')$ corresponding to some vertex w' of G', because $G(w')$ is 2-edge connected. Assume that $u \in G(u')$ and $v \in G(v')$ for two distinct nodes $u', v' \in V(G')$. We know that G' is 2-edge connected, and so there are two edge-disjoint simple paths between u' and v' in G'. We extend these paths into two edge-disjoint path between u and v in G such that, whenever a path in G' visits a vertex $w' \in V(G')$, we replace w' with a path in $G(w')$ between the incident edges of $E(G')$. If only one path visits a node $w' \in V(G')$, then such a path exists because $G(w')$ is connected. If both paths visit w', then we connect two pairs of vertices in $G(w')$ by edge-disjoint paths, which can be done because $G(w')$ is 2-edge connected. $\square$

3 Partition Algorithm

We describe how to choose a subset $S' \subset S$ and generate a convex partition for S'. Intuitively, we grow a *separator*, which consists of input segments and some of their extensions. We stop when every segment in the separator has been extended in both directions. At that time, each face of the separator is convex. We start with a segment having an endpoint along the convex hull $\mathrm{conv}(\cup S)$ and extend it beyond the other endpoint. If an extension $\overrightarrow{r}$ hits a new segment, we include that segment into S' and extend it beyond its endpoint that is in counterclockwise position to the ray $\overrightarrow{r}$. If an extension hits the separator or infinity, then we next consider a segment in the separator that has been extended in only one direction, and now extend it in the opposite direction. We choose the segment with maximal *turning angle*, where the turning angle of a directed segment in the separator is the total counterclockwise angle along the path from the initial segment. To define the turning angle, we maintain the invariant:

Invariant 1 *There is a segment $s_0 \in S'$ incident to $\mathrm{conv}(\cup S)$ such that, for every segment $t \in S'$, there is a chain of segments $(s_0, s_1, \ldots, s_k = t)$, $k \in \mathbb{N}$, in S' and an extension $\overrightarrow{r_i}$ of s_i hits s_{i+1} for $i = 0, 1, \ldots, k-1$.*

If a ray $\overrightarrow{r}$ hits a directed segment $\overrightarrow{s}$, then their turning angle $\angle(\overrightarrow{r}, \overrightarrow{s}) \in (-\pi, \pi)$ is the counterclockwise angle of rotation that carries $\overrightarrow{r}$ to $\overrightarrow{s}$. For a directed segment $\overrightarrow{t}$, $s_i \in S'$, a chain $(s_0, s_1, \ldots, s_k = t)$ determines a *total turning angle* as $\gamma = \sum_{i=1}^{k} \angle(\overrightarrow{s_{i-1}}, \overrightarrow{s_i})$. We can now present our partition algorithm.

Partition Algorithm. Input S.

- Pick a segment $s_0 = a_0b_0$ with an endpoint b_0 along $\mathrm{conv}(\cup S)$. Set $s := s_0$, $p := a_0$, $\gamma := 0$, $S' := \emptyset$, and $i := 1$.
- Repeat while $p \neq b_0$:

  Extend s beyond p into a ray $\overrightarrow{r}$ until it hits another segment, a previous extension, or to infinity.

  - If $\overrightarrow{r}$ hits a segment in $S \setminus S'$, denote it by $s_i = a_ib_i$ such that $\angle(\overrightarrow{r}, \overrightarrow{a_ib_i}) < 0 < \angle(\overrightarrow{r}, \overrightarrow{b_ia_i})$, let $\gamma_i = \gamma + \angle(\overrightarrow{r}, \overrightarrow{a_ib_i})$, put $S' := S' \cup \{s_i\}$, $s := s_i$, $p := a_i$, $\gamma := \gamma_i + \pi$, and $i := i + 1$.
  - Else, over all integers j, $0 \leq j < i$, such that $s_j \in S'$ has not been extended beyond b_j, pick one where the turning angle γ_j is maximal. Set $s := s_j$, $p := b_j$, and $\gamma := \gamma_j$.

It is immediate that the segments S' selected by our algorithm and their extensions form a convex partition P' of S' such that every segment of $S \setminus S'$ lies in the interior of a face of P'. It is also clear that Invariant 1 is maintained. It remains to show that the dual graph G' of P' is 2-edge connected.

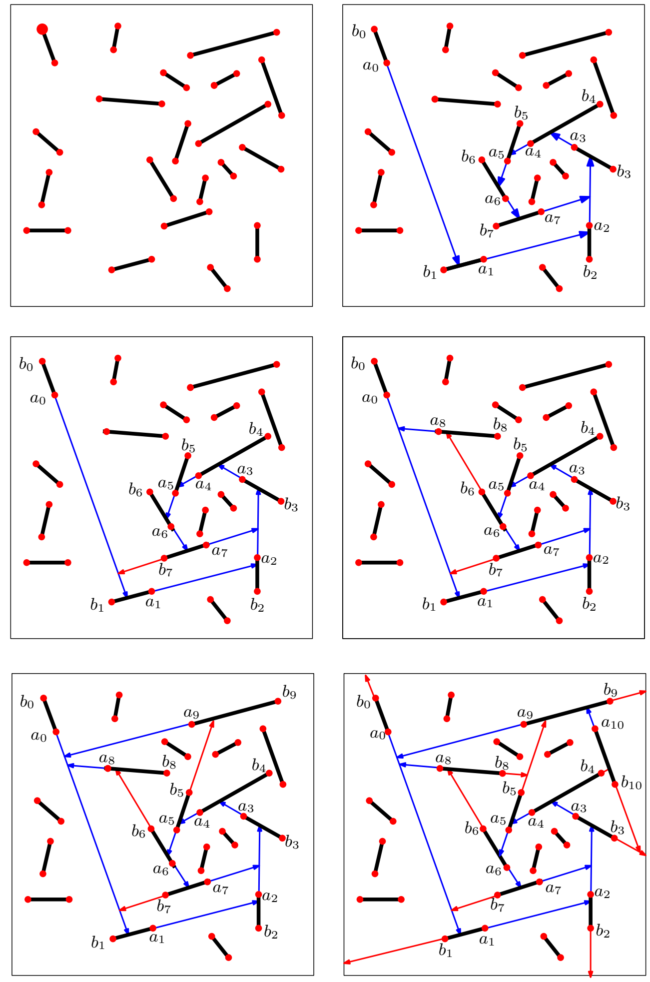

Figure 4: Steps of our partition algorithm.

4 The Dual Graph of P' is 2-Edge Connected

Let $\mathcal{R}$ denote the set of all simply connected regions that are unions of some, but not all, convex faces of P'. It is enough to show that every region $R \in \mathcal{R}$ has at least two segment endpoints on its boundary ∂R. We *direct* each extension in the natural direction such that they emanate from the corresponding segment endpoint. The input segments, however, are not directed. The boundary of each region $R \in \mathcal{R}$ contains portions of some nondirected input segments and directed extensions.

Lemma 3 *For every $R \in \mathcal{R}$, if ∂R contains both clockwise and counterclockwise extensions, then ∂R contains an entire input segment.*

Proof. The starting point of every directed portion of ∂R is a segment endpoint. The shortest portion $\gamma \subset \partial R$ between clockwise and counterclockwise extensions must be incident to two segment endpoints. Because the segments are disjoint, $\gamma \subset \partial R$ is a segment. $\square$

In particular, the boundary of every region $R \in \mathcal{R}$ that extends to infinity contains an entire segment.

Lemma 4 *For every $R \in \mathcal{R}$, if ∂R contains only clockwise (resp., counterclockwise) extensions, then ∂R contains at least two segment endpoints.*

Proof. Assume to the contrary that there is a region $R \in \mathcal{R}$ whose boundary contains fewer than two segment endpoints. Our algorithm inserts extensions one by one, so there is a linear order of the extensions along ∂R. Each extension has a segment endpoint at its tail.

In each connected portion of ∂R that consists of extensions only, the youngest extension must lie entirely in ∂R, hence the segment endpoint at its tail also lies in ∂R. This proves that there is at least one endpoint p of some segment s along ∂R. Furthermore, if p is the only segment endpoint along ∂R, then ∂R consists of a connected portion of s and a connected arc of extensions.

Note that ∂R cannot be a path extending to infinity, because then the extensions on the two extremes would have opposite orientations with respect to R. Hence ∂R is a circuit. We may assume without loss of generality that R lies in the interior of ∂R. It follows that R is a polygon, having at least three sides, and so at least three extensions lie along ∂R. Every extension along ∂R hits either s or a previous extension. By Lemma 4, all extensions along ∂R have the same orientation. This gives a complete order in which these extensions are created in our partition algorithm: first an extension $\overrightarrow{r_0}$ that hits s, and last the extension $\overrightarrow{r}$ of s beyond p.

Extensions do not cross each other, so a line segment in the interior of R cannot participate in a chain described in Invariant 1. Hence the segments whose extensions lie along ∂R cannot be in the interior of R. It follows that R is convex: it cannot have reflex angles, because each vertex is the intersection of extensions of segments that do not lie in its interior. We are left with two cases:

Case 1: all extensions along ∂R are counterclockwise. In our algorithm, after $\overrightarrow{r_0}$ hits s, the next step extends s beyond p. This contradicts our finding that another extension along ∂R is drawn between $\overrightarrow{r_0}$ and $\overrightarrow{r}$.

Case 2: all extensions along ∂R are clockwise. This also contradicts the progress of our algorithm. After ray $\overrightarrow{r_0}$ hits segment s, no extension can hit $\overrightarrow{r_0}$ from the right before the extension $\overrightarrow{r}$ of s is drawn, because any ray hitting $\overrightarrow{r_0}$ from the right must have a strictly smaller turning angle than that of $\overrightarrow{r}$. □

The runtime of our algorithm is dominated by the $2n$ ray shooting queries, one for each segment endpoint, among $O(n)$ noncrossing segment obstacles. Note that the set of obstacles dynamically increases as the new extensions are drawn. Using the currently known best dynamic data structures, $2n$ ray shooting queries and $2n$ segment insertions can be done in $O((n^2/\sqrt{k})\log^{O(1)} n)$ time using $O(k^{1+\varepsilon})$ space and preprocessing, where $n \le k \le n^2$ is a parameter and $\varepsilon > 0$ is an arbitrarily small positive constant [1, 2, 4].

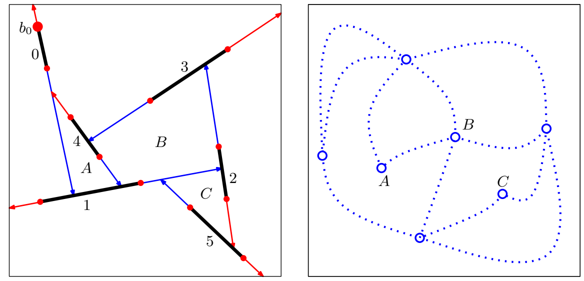

Figure 5: The boundary of every region in $\mathcal{R}$ either contains an entire segment (e.g., face A) or contains extensions of the same orientation (B, C, and $A \cup B$).

With $k = n^{4/3}$, our algorithm can be implemented in $O(n^{4/3+\varepsilon})$ time and space; with $k = n$, it requires $O(n^{1+\varepsilon})$ space and $O(n^{3/2} \operatorname{polylog} n)$ time.

References

[1] P. K. Agarwal and J. Erickson, Geometric range searching and its relatives, in *Advances in Discrete and Computational Geometry*, vol. 223 of Contemporary Mathematics, AMS, 1999, pp. 1-56.

[2] P. K. Agarwal and M. Sharir, Applications of a new space partitioning technique, *Discrete Comput. Geom.* **9** (1) (1993), 11–38.

[3] O. Aichholzer, S. Bereg, A. Dumitrescu, A. García, C. Huemer, F. Hurtado, M. Kano, A. Márquez, S. Smorodinsky, D. Souvaine, J. Urrutia, and D. R. Wood, Compatible geometric matchings, in preparation.

[4] S. W. Cheng and R. Janardan, Algorithms for ray-shooting and intersection searching, *J. Algorithms* **13** (1992), 670–692.

[5] M. Hoffmann, B. Speckmann, and Cs. D. Tóth, Pointed binary encompassing trees: simple and optimal, *Comput. Geom. Theory Appl.* (2007), to appear.

[6] Z. Jin, M. Kano, X. Li, and B. Wei, Partitioning 2-edge-colored complete multipartite graphs into monochromatic cycles, paths and trees, *J. Comb. Optim.* **11** (4) (2006), 445–454.

[7] A. Kaneko and M. Kano, Discrete geometry on red and blue points in the plane—a survey, in *Discrete and computational geometry*, vol. 25 of Algorithms Combin., Springer, Berlin, 2003, pp. 551–570.

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

[9] Cs. D. Tóth, Illumination in the presence of opaque line segments in the plane, *Comput. Geom. Theory Appl.* **21** (3) (2002), 193–204.

[10] Cs. D. Tóth, Illuminating disjoint line segments in the plane, *Discrete Comput. Geom.* **30** (3) (2003), 489–505.

[11] J. Urrutia, Art gallery and illumination problems, in *Handbook of Computational Geometry (Eds. J.-R. Sack and J. Urrutia)*, Amsterdam, North-Holland, 2000, pp. 973–1027.