Properties of Arrangement Graphs *

P. Bose† H. Everett‡ S. K. Wismath§

Abstract

An arrangement graph G is the abstract graph obtained from an arrangement of lines L, in general position by associating vertices of G with the intersection points of L, and the edges of G with the line segments joining the intersection points of L. A simple polygon (respectively path) of n sides in general position, **induces** a set of n lines by extension of the line segments into lines. The main results of this paper are:

- Given a graph G, it is NP-Hard to determine if G is the arrangement graph of some set of lines.
- There are non-Hamiltonian arrangement graphs for arrangements of six lines and for odd values of $n > 6$ lines.
- All arrangements of n lines contain a subarrangement of size $\sqrt{n-1}+1$ with an inducing polygon.
- All arrangements on n lines contain an inducing path consisting of n line segments. A Java applet implementing the algorithm for determining such a path is also provided.
- All arrangements on n hyperplanes in R^d contain a simple inducing polygonal cycle of size n.

1 Introduction

A finite set L of lines in R^2 partitions the plane into connected pieces of dimensions 0, 1, 2, called an **arrangement**, denoted as $\mathcal{A}(L)$. If the lines are in general position, i.e. if no pair is parallel and no triple of lines intersect at a common point, then the arrangement is said to be **simple** (or **uniform**), and there are clearly $\binom{n}{2}$ points of intersection. Arrangements of lines have been studied extensively in a variety of contexts and there is a large literature. The text by Edelsbrunner [1] provides a good introduction to the area; other related references include [2] and [5].

A (line) **arrangement graph** (denoted as $G(L)$) is the abstract graph obtained from $\mathcal{A}(L)$ where the vertex set of $G(L)$ is the $\binom{n}{2}$ intersection points of $\mathcal{A}(L)$ and two vertices v_1 and v_2 are adjacent in $G(L)$ if and only if the intersection points corresponding to v_1 and v_2 share a common line of L and the line segment of finite length joining them intersects no other line of L (i.e. they are consecutive on some line of L). Since each line intersects $n-1$ lines, there are exactly $n(n-2)$ edges in $G(L)$. This graph is planar and furthermore the degree of each vertex is either $2, 3$, or 4. Although many properties of arrangements are known, only a few papers have explicitly dealt

*The authors gratefully thank N.S.E.R.C. for financial assistance.

†Department of Computer Science, Carleton University, Ottawa, Canada. `http://www.scs.carleton/~jit`

‡Département d'Informatique, Université du Québec à Montréal, Montréal, Canada. `http://www.info.uqam.ca/~everett`

§Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, Canada. `http://www.cs.uleth.ca/dept/wismath.html`

with the arrangement *graph*, in particular for the case in which an embedding of the graph is not specified. In Eu et al. [4], an efficient algorithm for computing the envelope, or outer face of the (embedded) arrangement graph was presented, extending an earlier result by Keil [7].

In this paper, we investigate the following properties of arrangement graphs:

- Can they be recognized efficiently?
- Are they Hamiltonian?
- Do they contain a simple inducing polygon?

2 Arrangement Graph Recognition

The **Arrangement Graph Recognition** problem is to determine if a given graph G is the arrangement graph of some (unknown) set of lines. In this section, we show that the problem is NP-Hard. The reduction is from the Pseudo-line Stretchability problem to be discussed later in this section. If an *embedding* of G were specified, then the reduction would be a straight-forward reformulation. However, the proof is not so direct since a given arrangement graph may admit many different planar embeddings. First we consider some necessary properties of arrangement graphs.

If G is an arrangement graph with v vertices, then the following properties hold:

1. All vertices of G have degree 2, 3, or 4.

2. There exists an embedding of G with all vertices of degree 2 or 3 on the exterior face. Call the resulting graph Γ.

3. Let n represent the number of lines in the arrangement. Then $v = \binom{n}{2}$, or alternately, $n = \frac{1+\sqrt{1+8v}}{2}$. Then Γ can be decomposed into n edge disjoint paths, corresponding to the lines of the underlying arrangement, each of length $n-1$ such that:

   - each path starts and stops at a vertex of degree 2 or 3
   - each degree 2 vertex is an endpoint of exactly two paths
   - each degree 3 vertex is an endpoint of exactly one path (along the unique edge bordering two *interior* faces of Γ).
   - each degree 4 vertex w is on exactly two paths that cross at w.

We now show how a unique embedding can be determined from a given abstract graph G. Let G^* be the graph obtained by adding one new vertex x to G, connected to each vertex of degree 2 or 3. See Figure 1.

Lemma 2.1. *If G is an arrangement graph, then G is biconnected.*

Proof. Consider an arbitrary pair of vertices u and v in G. In the arrangement, u and v are each incident on two lines; consider the sub-arrangement induced by the lines containing u and v. Note that, as shown by the examples in Figure 2, this sub-arrangement consists of either three or four lines. It is easy to check that there are two vertex disjoint paths between any pair of vertices u and v.

□

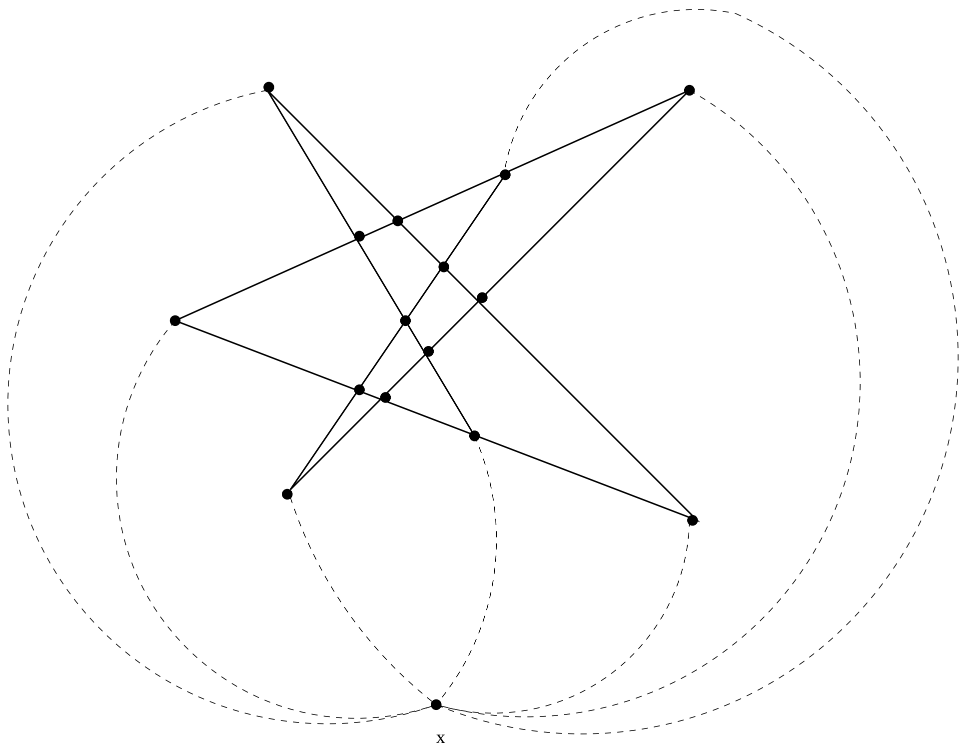

Figure 1: G^*

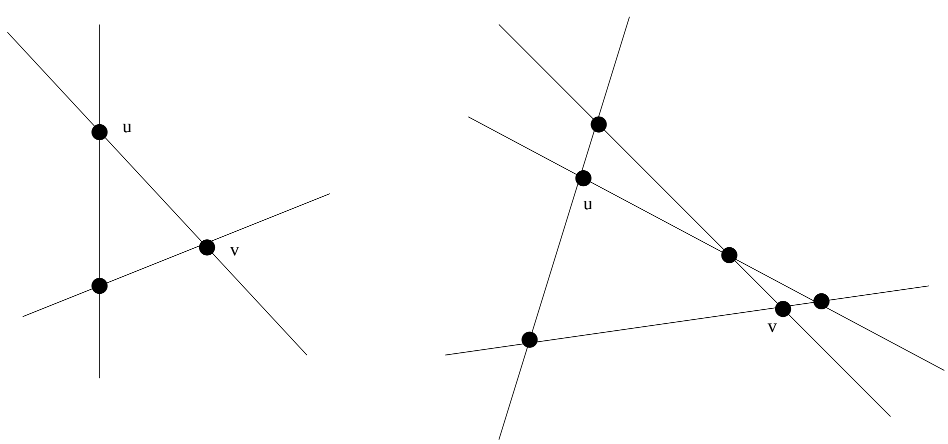

Figure 2: Induced Sub-arrangements

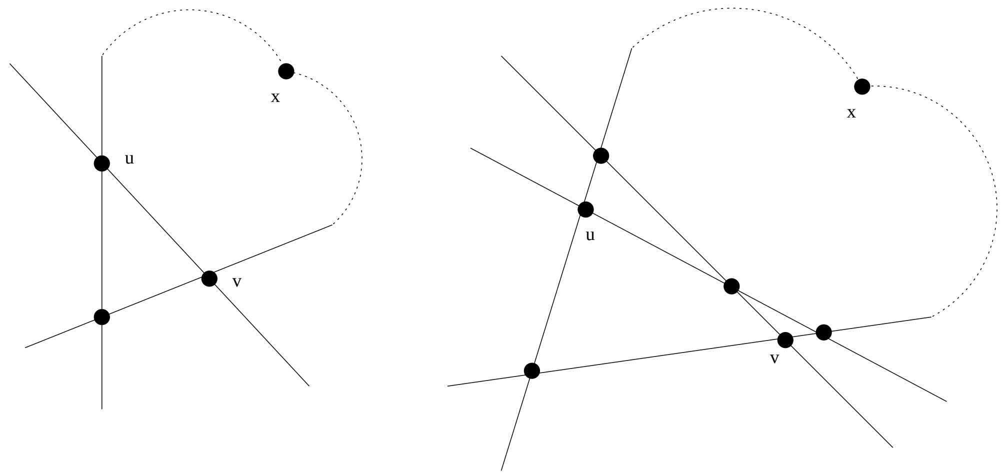

Figure 3: Sub-arrangements in G^*

Lemma 2.2. *If G is an arrangement graph, then G^* is triconnected.*

Proof. Let u and v denote an arbitrary pair of vertices in G^*. If neither u nor v is vertex x, then consider the sub-arrangement induced on u and v as in Figure 3. Let P_1 and P_2 be two vertex disjoint paths between u and v in this subarrangement. The vertices on the envelope of the sub-arrangement are connected to vertex x and hence u and v are each connected by at least one path to x, either directly if they are on the envelope of G, or by a path of vertices not intersecting P_1 or P_2, and thus three vertex disjoint paths between u and v can be determined. A similar argument applies if x is chosen as one of the vertices.

□

The graph G^* is planar by construction and by lemma 2.2, it is triconnected. As a consequence of a result due to Whitney [10], there exists a unique embedding of G^* in the plane with x on the exterior face. Let Γ be this unique embedding of G^* with x and incident edges removed. If G is an arrangement graph, then it is easy to see that Γ, can be decomposed in polynomial time into a set of n edge disjoint paths corresponding to the lines as noted in property c.

A **pseudo-line** is a generalization of a line, defined as a curve in the plane that does not self intersect and that goes to infinity in two directions. A pseudo-line arrangement is a collection of n pseudo-lines with the property that each pair intersects at most once and cross if they intersect. As for line arrangements, simple (or uniform) pseudo-line arrangements have the property that each pair of pseudo-lines intersect exactly once and no triple intersects at a point. Some pseudo-line arrangements are combinatorially equivalent to straight line arrangements, in which case the pseudo-line arrangement is said to be **stretchable**. It is known that all sets of eight or fewer pseudo-lines are stretchable and as a consequence of Pappus' Theorem, there exists a pseudo-line arrangement of nine lines that is not stretchable. Shor [8] showed that the Pseudoline Stretchability problem is in fact, NP-Hard.

Theorem 2.1. *Arrangement Graph Recognition is NP-Hard*

Proof. The reduction is from Pseudo-line Stretchability. Given an instance of the Pseudo-line Stretchability problem, create a Pseudo-Arrangement Graph G. G satisfies the necessary properties of arrangement graphs, however, G is stretchable if and only if G is an arrangement graph.

□

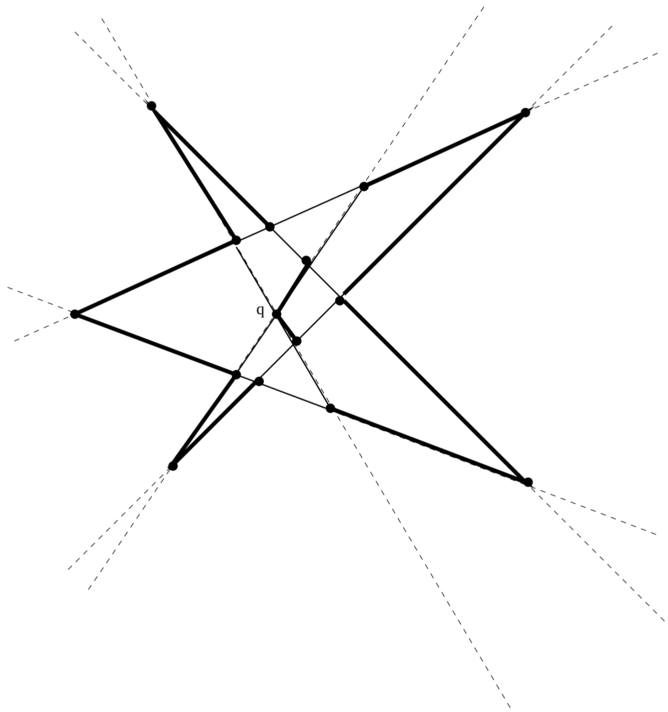

Figure 4: No Hamiltonian cycle

The recognition of arrangement graphs is NP-Hard, however knowledge of which vertices lie on each line is computable by a simple greedy algorithm in linear time. In [11], Wismath presents a polynomial-time algorithm for reconstructing a set of points from their order type information *given their x-coordinates*. Via a duality transformation, this is equivalent to reconstruction of a set of lines given the ordering of the line incidences along each line *together with the slopes of each line*. As a consequence of this result, and in contrast to the previous theorem, arrangement graphs can be recognized efficiently, given sufficient extra information.

Corollary 2.1. *Arrangement Graph Recognition is in P if the slope of each line is specified.*

The same algorithm reconstructs lines (and hence recognizes arrangement graphs) if the X-intercepts are specified instead of slopes.

3 Hamiltonicity

One natural question is whether arrangement graphs are necessarily Hamiltonian. Hazel Everett was the first to show that not all arrangement graphs admit a Hamiltonian cycle. The arrangement graph in Figure 4 is the smallest graph (six lines) that does not contain a Hamiltonian cycle. It can be verified by inspection that all arrangements on fewer than six lines contain a Hamiltonian cycle. That the arrangement graph of Figure 4 does not contain a Hamiltonian cycle follows from the observation that any Hamiltonian cycle must follow the highlighted paths through the degree two vertices. Now consider the vertex labelled q. Since q is adjacent to two vertices that are already forced, the Hamiltonian cycle is forced to pass through q via the other two neighbours of q, as indicated. There is no way to join the three portions of the Hamiltonian cycle together.

Theorem 3.1. *There exist arrangements of n lines that do not contain a Hamiltonian cycle, for $n = 6$ and all odd values of $n > 6$.*

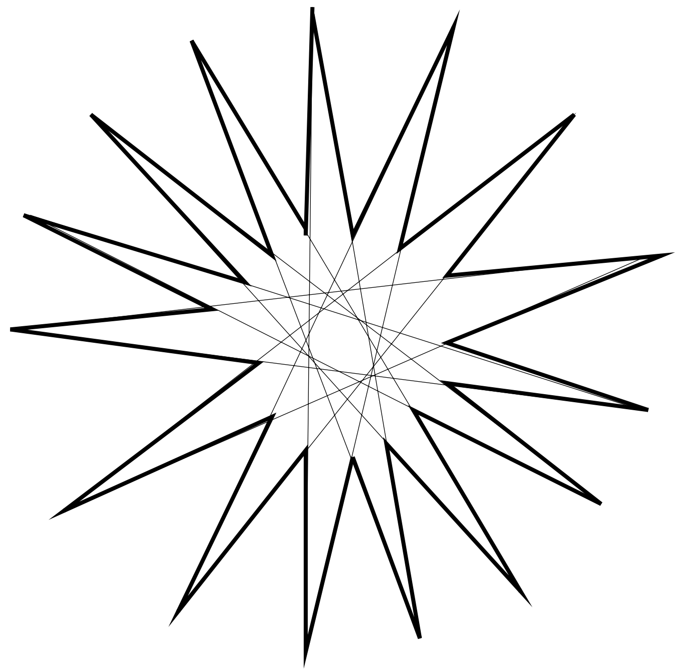

Figure 5: No Hamiltonian cycle for odd values of n

Figure 5 shows the general construction for fifteen lines. Note that any Hamiltonian cycle must traverse the outer cycle of vertices and is hence unable to include any interior vertex.

Great circle arrangement graphs can be defined as the graphs obtained by stereographically projecting a set of n great circles of a sphere onto the plane. The graph consists of the $2\binom{n}{2}$ intersection points as vertices and $n \cdot 2(n-1)$ edges corresponding to pairs of intersection points that are adjacent on some circle. Such graphs are planar and 4-connected and hence always contain a Hamiltonian cycle, by a result due to Tutte [9]. Iwamoto and Toussaint [6] defined a related class of arrangement graphs called Jordan curve arrangement graphs and showed that determining if such a graph contains a Hamiltonian cycle is an NP-Complete problem.

4 Inducing Polygons

A simple polygon of n sides in general position **induces** a set of n lines by extension of the sides into *lines*. In this section, we consider the converse problem - refer to figure 6.

Induced Polygon in Arrangement (IPA): Given a set L of n lines, does there exist a *simple* n-gon that induces L?

For the remainder of this section, let $L = \{l_1, l_2, ...l_n\}$ be a set of lines in the plane indexed in decreasing order of slope. We assume the lines are in general position - *i.e.* no pair of lines are parallel and no triple of lines intersect at a common point. In general, an inducing n-gon need not be unique for a given set of lines. Although it remains an open problem to determine if all arrangements contain an inducing polygon, it is possible to exhibit classes of arrangements that are guaranteed to contain an inducing polygon. The k-sail arrangements (for k = 0,1,2,3) are families of arrangements introduced by Eu, Guévremont, and Toussaint [4]; the inducing polygon for any k-sail arrangement A is exactly the *envelope* of A. In the same paper, the authors introduce the notion of 1-exterior arrangements which can be shown to be precisely the set of arrangements for which the envelope forms an inducing polygon.

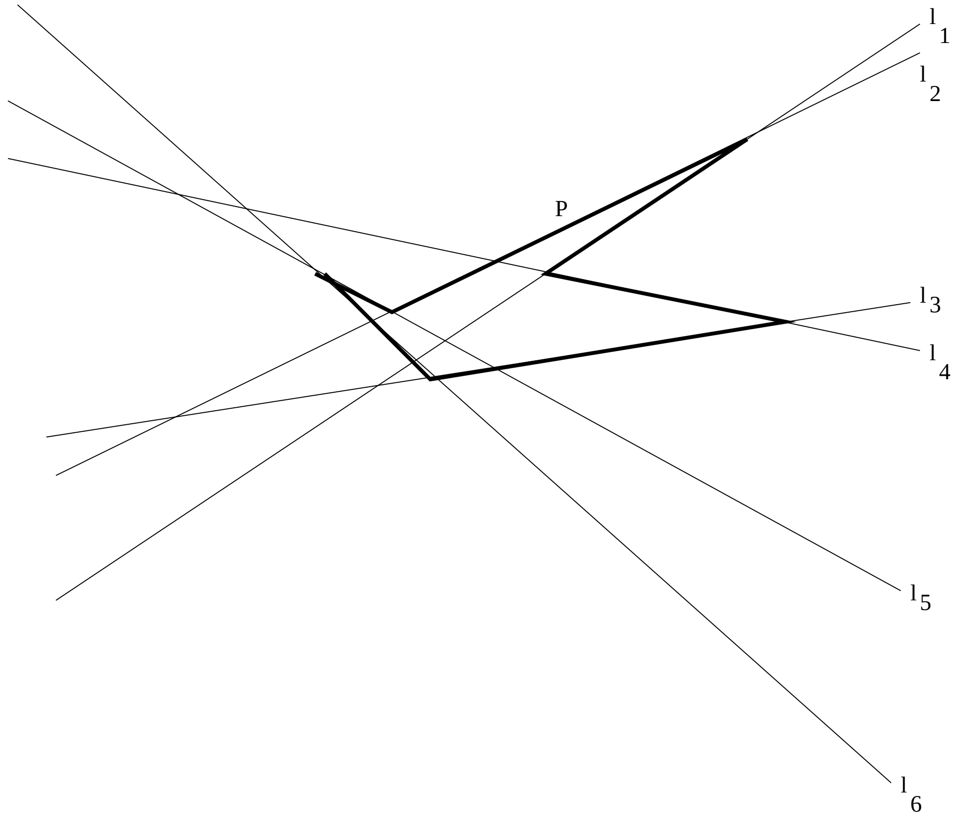

Figure 6: Inducing polygon

4.1 Inducing n-paths

If the Inducing Polygon problem is weakened slightly to determining a simple inducing *path* of n sides (also known as an n-path or poly-line) then all arrangements can be shown to be inducable. Here we assume that all corners of an inducing n-path, including the first and last corners, must appear at intersection points of the induced arrangement. Furthermore, we allow the degenerate case in which the n-path forms an n-gon, *i.e* when the first and last corners of the path are identical.
Induced n-Path in Arrangement (IpA): Given a set L of n lines, does there exist a *simple* n-path that induces L?

The main result of this section is an affirmative answer to the IpA problem.

Theorem 4.1. *Every arrangement of n lines contains a simple inducing n-path that can be determined in $O(n^2)$ time and $O(n)$ space.*

Proof. We first provide an algorithm that produces an inducing $(n-1)$-path that starts and finishes on the unused line. The algorithm is most easily described in terms of a point tracing a simple path through the arrangement - only the points at which the path turns are output. Initially all lines are labelled as "unused" and a point of intersection on the envelope is determined as a starting point. Each time an unused line is encountered, it is immediately used by turning left on it if possible and right otherwise. More precisely:

- `Mark all lines as` *unused*`.`
- `Determine the rightmost intersection point` s `on line` l_1 `with a line` l_i`. Output` s`.`
- `Mark` l_1 `as` *used*`.`
- `At` s `face towards the next intersection point along` l_1`.`

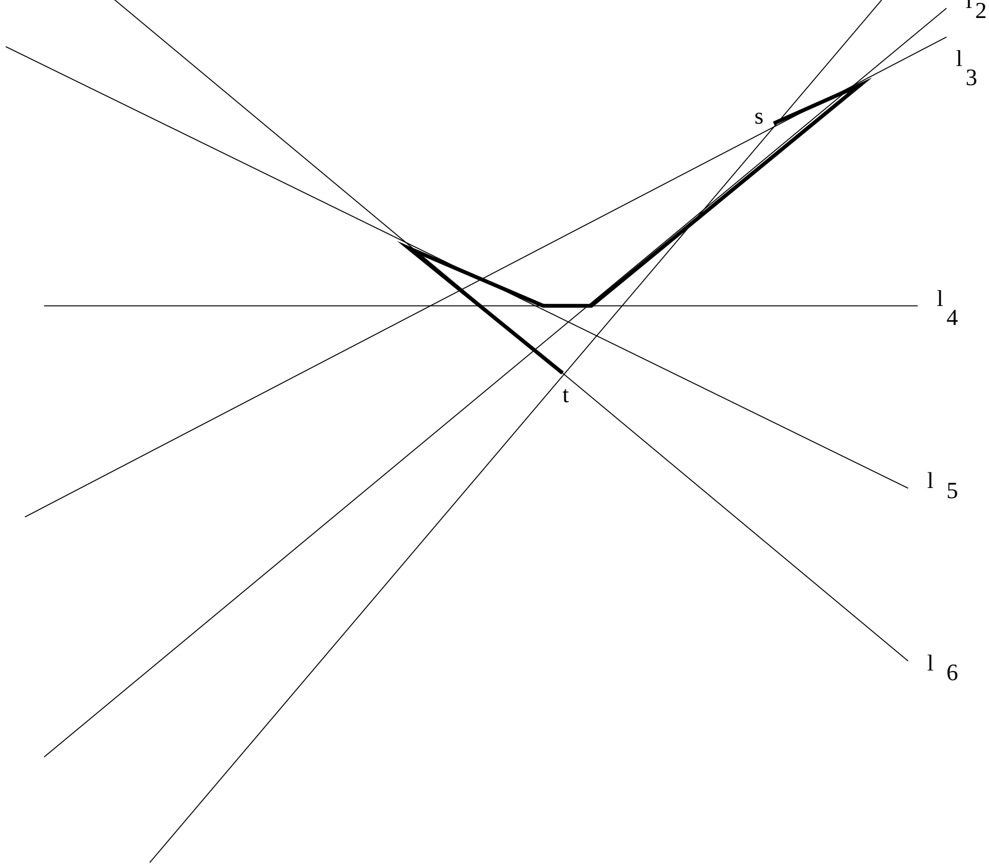

Figure 7: The algorithm performing on a set of lines

- `If` i `= 2 make a right turn on` l_2 `else turn left on line` l_i.
- `Mark line` l_i `as` *used*.
- *current* $:= i$.
- `While there are` *unused* `lines do`
  - `Proceed along line` $l_{current}$ `to the first` *unused* `line` l_i. `Output` $l_{current} \cap l_i$.
  - `Mark` l_i `as` *used*.
  - `If there are` *unused* `lines with index` $< i$ `then turn left onto line` l_i `else turn right onto` l_i.
  - *current* $:= i$
- `Proceed to` $t = l_{current} \cap l_1$. `Output` t.

Figure 7 shows the path traced by the algorithm and the resulting $(n-1)$-path. A Java applet implementing the algorithm is available at `http://www.cs.uleth.ca/~wismath/arrange.html`. The correctness of the algorithm follows from three simple observations:

- Each line of the arrangement intersects all other lines and hence each time a new unused line is chosen as $l_{current}$, it intersects all unused lines.
- The path created does not cross itself. Whenever an unused line is encountered, it is immediately chosen. If the path intersected itself at a point p, when proceeding along a line l_i, then the algorithm must have chosen l_i when it passed through point p and marked it as used.

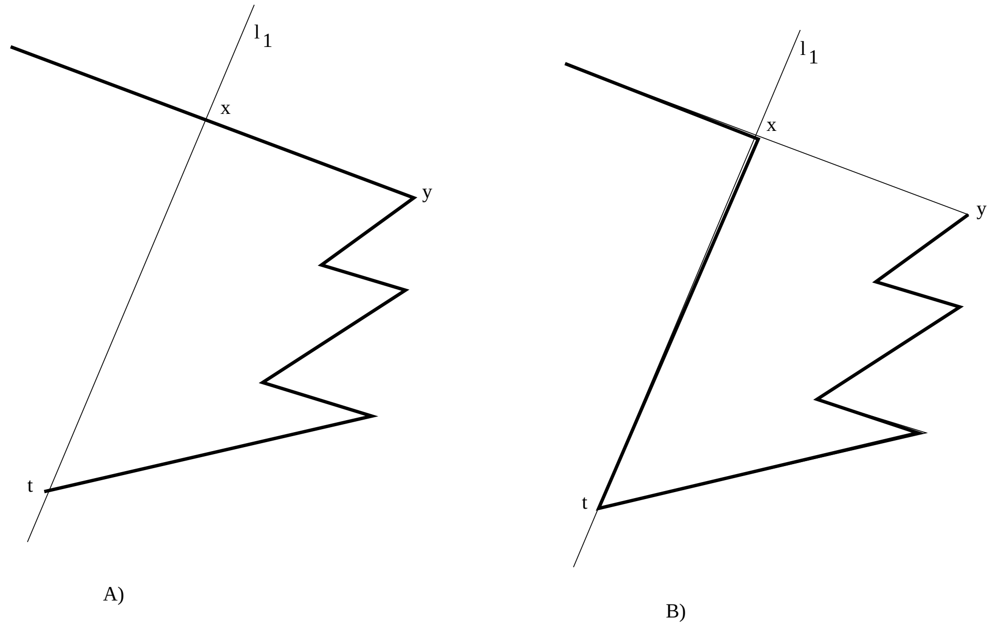

Figure 8: Revising the Path

- When an unused line l_i is encountered, if there are unused lines to the right on l_i, they must have an index less than i (the algorithm turns *left* to reach them). An unused line with index greater than i that intersects l_i to the right, must necessarily cross the path - contradicting the assumption that it is unused.

Finally, note that the last leg of the path to l_1 can not intersect the path, otherwise the final unused line would have been chosen earlier. The resulting path has sides on all lines other than l_1 and hence is a simple $(n-1)$-path and has both endpoints on l_1 - the unused line.

The $(n-1)$-path produced by the algorithm can be modified into a simple n-path. When l_1 is encountered in the final step, make a turn towards the starting point s on l_1 to the first encountered intersection point x. Either x is not on the path formed by the $(n-1)$-path, in which case the n-path is obtained directly, or this final step has created a single cycle as in Figure 8A which can be revised as in 8B to create a valid simple inducing n-path.

Analysis: A straight-forward brute force implementation of the algorithm can be accomplished in $O(n^2)$ time and $O(n)$ space. □

4.2 Consequences

The correctness argument still applies if the algorithm turns in an *arbitrary* direction at each stage, as long as this turn results in an encounter with some unused line. Additionally, the starting point may be chosen arbitrarily, with only minor modifications to the algorithm - in particular, checking indices no longer insures successful turns. It is also clear that the three observations also apply for *pseudo-lines* and thus the algorithm can be modified to find a simple inducing n-pseudo-path in a set of n pseudo-lines.

Note that if all intersections lie on one side of l_1, then the $(n-1)$-path produced can be converted to a simple n-gon by simply joining the starting and finishing points. It can easily be shown that all intersections lie on one side of l_1 if and only if the lines $l_2, ..., l_n$ cross l_1 *in sorted order* by index. Although in general this condition may not hold for any line of the arrangement, the following lemma due to Erdös and Szekeres [3] will be used to ensure the existence of a subset of interesting lines.

Lemma 4.1. *(Erdös-Szekeres) In any sequence of* n *distinct integers, there exists a subsequence of at least* $\sqrt{n}$ *integers that is either strictly increasing or strictly decreasing.*

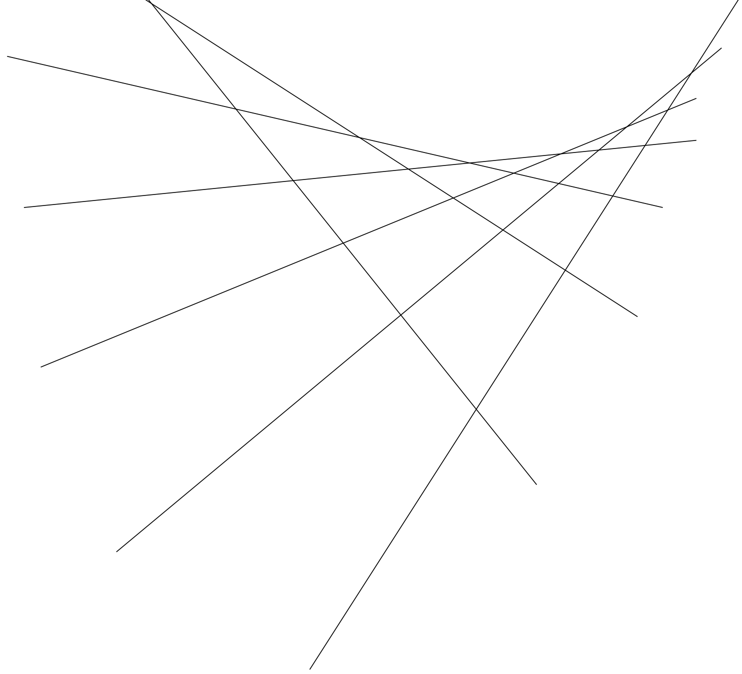

Figure 9: No convex k-gon for $k > 4$

Consequently, it is possible to determine a subarrangement of size $O(\sqrt{n})$ that contains a simple inducing polygon.

Theorem 4.2. *In any arrangement L of n lines, there exists a subset L' of p lines such that L' contains a simple inducing p-gon, for all values of p with $3 \leq p \leq \sqrt{n-1}+1$.*

Proof. Consider the sequence of the $n-1$ indices of the lines crossing line l_1. By the previous lemma, there is a subsequence of indices of size $\sqrt{n-1}$ that is strictly increasing or decreasing. This subarrangement has the property that all intersection points lie in one of the closed half planes defined by l_1 and hence the previous algorithm yields a $\sqrt{n-1}$-path that can be safely extended to a simple inducing $\sqrt{n-1}+1$-gon. □

Note that by renumbering the lines, any line may be chosen as the initial line and thus can be chosen to be in the p-gon.

Joining the starting and finishing points s and t could also be safely accomplished if line l_1 is allowed to wrap around through the point at infinity - a notion better expressed in terms of great circles.

Corollary 4.1. *All arrangements of n great circles (in general position) on a sphere contain an inducing n-gon.*

Proof. Given a set of n great circles on a sphere S with centre c, project the circles onto a tangent plane P from c producing a set of n lines. Consider the image of the $(n-1)$-path created by the algorithm on the circles on S. The single unused circle corresponding to l_1 can be joined to the $(n-1)$-path via the long arc between the images of the points s and t. □

Another interesting variant of the Inducing Simple Polygon problem is to identify a particular type of simple sub-polygon in a given arrangement, for example by specifying convex/reflex constraints of the vertices. David Bremner has noted that there is a family of arrangements that contain no convex k-gon for $k > 4$, as shown in figure 9. It can also be seen that if an arrangement contains a *convex* inducing n-gon C, then C must appear as an interior face in the arrangement.

4.3 Inducing Polygons in Higher Dimensions

In d dimensions, a finite set of hyperplanes H (i.e. $(d-1)$-dimensional flats) defines an arrangement (denoted $\mathcal{A}(H)$) by partitioning R^d into connected pieces of dimensions 0, 1, $\ldots, d$. If the hyperplanes are in general position, that is, no $d+1$ subset of hyperplanes intersect at a common point, then the arrangement is called *simple*. Edelsbrunner [1] provides a good introduction to the area; other related references include [2] and [5].

Recall that a polygon P is an inducing polygon of a line arrangement $A(L)$ provided it has the following properties:

1. P is a simple polygon,
2. every vertex of P is a vertex of $A(L)$,
3. every edge of P lies on a line of L,
4. each line in L is defined by 2 consecutive vertices of P.

The notion of an inducing polygon can be generalized to arrangements in higher dimensions. We define a polygonal cycle in d dimensions to consist of a sequence of n points $S = \{s_0, s_1, \ldots, s_{n-1}\}$ along with all edges $\overline{s_i s_{i+1}}$[1]. The polygonal cycle is simple provided that no pair of non-consecutive edges intersects properly and consecutive edges intersect only at endpoints.

A simple polygonal cycle P is an inducing polygonal cycle of an arrangement $A(H)$ of hyperplanes in general position provided it has the following properties:

1. P is a simple polygonal cycle,
2. every vertex of P is a vertex of $A(H)$,
3. every edge of P lies on the intersection of $d-1$ hyperplanes of H,
4. each hyperplane in H is defined by d consecutive vertices of P.

Although we were unable to prove that every line arrangement contains an inducing simple polygon, with the extra freedom available in higher dimensions, we prove that every arrangement of hyperplanes in R^d $(d > 2)$ has a simple inducing polygonal cycle that can be computed in $O(n)$ time and space.

Theorem 4.3. *Every arrangement of n hyperplanes in R^d (for fixed $d > 2$) in general position contains a simple inducing polygonal cycle of size n that can be determined in $O(n)$ time and space.*

Proof. Let $H = \{H_0, H_1, \ldots, H_{n-1}\}$ be a set on n hyperplanes in R^d in general position and $A(H)$ the arrangement induced by H.

Consider the following polygonal cycle $I_P = (H_0 \cap H_1 \cap \ldots \cap H_{d-1}, H_1 \cap H_2 \cap \ldots \cap H_d, H_2 \cap H_3 \cap \ldots \cap H_{d+1}, \cdots, H_{n-d} \cap H_{n-d-1} \cap \ldots \cap H_{n-1}, H_{n-d-1} \cap H_{n-d-2} \cap \ldots \cap H_0, \cdots, H_{n-1} \cap H_0 \cap \ldots \cap H_{d-2}, H_0 \cap H_1 \cap \ldots \cap H_{d-1})$

The polygonal cycle I_P is inducing because every d consecutive vertices lies on exactly one hyperplane of H. I_P is simple because the planes of H are in general position, i.e. if two non-consecutive edges of I_P intersected, this would imply that more than d hyperplanes intersect at a point.

The time and space complexity is $O(n)$ because essentially any ordering of the hyperplanes suffices to produce a valid inducing polygon. □

[1] indices are modulo n

5 Conclusions and Open Problems

Different properties of arrangements and arrangement graphs have been investigated in this paper. In summary, the main results and open problems are:

- Arrangement graph recognition is an NP-Hard problem. How much extra information is required to make the problem tractable?
- Hamiltonicity: Not all arrangement graphs are Hamiltonian. Determining if all arrangement graphs contain a Hamiltonian *path* remains as an open problem.
- Inducing Polygons: All arrangements contain a simple inducing $(n-1)$-path as produced by the algorithm [2] described in section 4.1. All arrangements have a subset of $O(\sqrt{n})$ lines which contain a simple inducing polygon. Do all arrangements contain a simple inducing n-gon?
- All arrangements of n hyperplanes in $R^d, d > 2$ contain a simple inducing polygonal cycle of size n.

6 Acknowledgments

Thanks are due to Godfried Toussaint for suggesting the arrangement graph recognition problem, to Ileana Streinu for discussions of the inducing polygon problem, to Patrice Belleville for discussions of the Hamiltonian cycle result, and to David Bremner for figure 9. The Java applet was written by J. Suwanparisut.

References

[1] H. Edelsbrunner, Algorithms in Combinatorial Geometry, Springer Verlag Publishers, 1987.

[2] H. Edelsbrunner, J. O'Rourke, R. Seidel, Constructing Arrangements of Lines and Hyperplanes with Applications, *SIAM J. Computing* **15** (1986), 341-363.

[3] P. Erdös, G. Szekeres, A Combinatorial Problem in Geometry, Composition Math. **2** (1935), 463-470.

[4] D. Eu, E. Guevremont, G. Toussaint, On Envelopes of Arrangements of Lines, *J. of Algorithms*, **21** (1996) 111-148.

[5] J. Goodman, R. Pollack, Multi-dimensional Sorting, *SIAM J. Computing* **12** (1983) 484-507.

[6] C. Iwamoto, G. Toussaint, Finding Hamiltonian Circuits in Arrangements of Jordan curves is NP-complete, *Information Processing Letters* **52** (1994) 183-189.

[7] M. Keil, A Simple Algorithm for Determining the Envelope of a set of Lines, *Information Processing Letters* **39** (1991), 121-124.

[8] P. W. Shor, Stretchability of Pseudolines is NP-Hard, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **4** (1991), 531-554.

[2] refer to `http://www.cs.uleth.ca/~wismath/arrange.html` for a Java implementation.

[9] W. T. Tutte, A Theorem on Planar Graphs, *Transactions American Math Society* **82** (1956), 99-116.

[10] H. Whitney, Congruent Graphs and the Connectivity of Graphs, *American J. of Mathematics*, LIV (1932), 150-168.

[11] S. K. Wismath, Point and Line Segment Reconstruction from Visibility Information, *International Journal of Computational Geometry*, **10(2)** (2000), 189-200.