# Acyclic Orientation of Drawings*

Eyal Ackerman† Kevin Buchin‡ Christian Knauer‡ Günter Rote‡


## Abstract

Given a set of curves in the plane or a topological graph, we ask for an orientation of the curves or edges which induces an acyclic orientation on the corresponding planar map. Depending on the maximum number of crossings on a curve or an edge, we provide algorithms and hardness proofs for this problem.


## 1 Introduction

Let $G$ be a *topological graph*, that is, a graph drawn in the plane such that its vertices are distinct points, and its edge set is a set of Jordan arcs, each connecting two vertices and containing no other vertex. In this work we further assume that $G$ is a *simple* topological graph, i.e., every pair of its edges intersect at most once, either at a common vertex or at a crossing point.

An *orientation* of (the edges of) a graph is an assignment of a direction to every edge in the graph. We say that an orientation is *acyclic* if the resulting directed graph does not contain a directed cycle. Finding an acyclic orientation of a given undirected (abstract) graph can be easily computed in linear time by performing a depth-first search on the graph and then orienting every *backward* edge from the ancestor to the descendent. However, is it always possible to find an orientation of the edges of a topological graph, such that a traveller on that graph will not be able to return to his starting position even if allowed to move from one edge to the other at their crossing point? Rephrasing it in a more formal way, let $M(G)$ be the planar map induced by $G$. That it, the map obtained by adding the crossing points of $G$ as vertices, and subdividing the edges of $G$ accordingly. Then we ask for an orientation of the edges of $G$ such that the induced directed planar map $M(G)$ is acyclic.

Clearly, if the topological graph is *x-monotone*, that is, every vertical line crosses every edge at most once, then one can orient each edge from its endpoint with the smaller $x$-coordinate towards its endpoint with the greater $x$-coordinate. Travelling on the graph under such orientation, one always increases the value of one's $x$-coordinate and therefore cannot form a directed cycle. However, not every topological graph is acyclic-orientable as Figure 1 demonstrates. Note that the degree of every vertex in this example is one. This gives rise for considering the orientation problem in the special case the degree of each vertex is one, or in other words, when one looks for an acyclic orientation of a set of *curves* embedded in the plane.

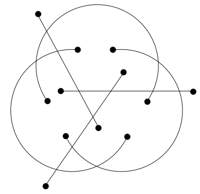


Figure 1: A non-orientable topological graph

It turns out that determining whether a topological graph (resp., a set of curves) has an acyclic orientation depends crucially on the maximum number of times an edge in the graph (resp., a curve) can be crossed. Given a (simple) topological graph $G$ on $n$ vertices, such that each edge in $G$ is crossed at most once, we show that one can find an acyclic orientation of $G$ in $O(n)$ time. When four crossings per edge are allowed, deciding whether there exists an acyclic orientation becomes NP-complete. Topological graphs with few crossings per edge were considered in several works in the literature [4, 5, 6]. For a set of $n$ curves in which each pair of curves intersects at most once and every curve is crossed at most $k$ times, we describe an $O(n)$-time orientation algorithm for the case $k \leq 3$. When $k \geq 5$ finding an acyclic orientation of the set of curves is NP-complete.

The rest of this paper is organized as follows. In Section 2 we study the problem of finding an acyclic orientation for a set of curves. Then, in Section 3 we consider the more general case where the input is a topological graph. Finally, we give some concluding remarks in Section 4, and mention a few related open problems.

## 2 Acyclic orientation of a set of curves in the plane

Throughout this paper we assume the intersections between the curves are known in advance. Given a set of curves $\mathcal{C}$, the vertices of the planar map $M(\mathcal{C})$, induced by $\mathcal{C}$, are the crossing points between the curves, while the edges of $M(\mathcal{C})$ are segments of the curves that connect two consecutive crossing points on a curve. As we have mentioned above, the maximum number of crossings per curve plays an impor-

*Work by the first author was done while he was visiting the Freie Universität Berlin, and was partly supported by a Marie Curie scholarship. Research by the second author was supported by the Deutsche Forschungsgemeinschaft within the European graduate program "Combinatorics, Geometry, and Computation" (No. GRK 588/2).

†Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel. ackerman@cs.technion.ac.il

‡Institute of Computer Science, Freie Universität Berlin, Takustr. 9, 14195 Berlin, Germany. {buchin|knauer|rote}@inf.fu-berlin.de

tant role when we ask for an acyclic orientation of a set of curves. If every curve is crossed at most once, then $M(\mathcal{C})$ contains no edges, and therefore any orientation of $\mathcal{C}$ is acyclic. If $\mathcal{C}$ is a set of curves with at most two crossing points per curve, then $M(\mathcal{C})$ is a union of cycles and paths and thus finding an acyclic orientation of $\mathcal{C}$ is also easy in this case. Hence, the first non-trivial case is where each curve is crossed at most three times. In this case we have:

**Theorem 1** *Let $\mathcal{C}$ be a set of $n$ curves in the plane, such that every pair of curves intersect at most once and each curve has at most three crossings. Then one can find an acyclic orientation of $\mathcal{C}$ in $O(n)$ time.*

This result is proved in Section 2.1, while in Section 2.2 we show:

**Theorem 2** *Let $\mathcal{C}$ be a set of curves in the plane, such that every pair of curves intersects at most once and each curve has at most five crossings. Then deciding whether $\mathcal{C}$ has an acyclic orientation is NP-complete.*

## 2.1 Curves with at most three crossings per curve

Let $\mathcal{C}$ be a set of $n$ curves in the plane, such that every pair of curves intersect at most once and each curve has at most three crossings. In this section we describe an algorithm for obtaining an acyclic orientation of $\mathcal{C}$. We start by constructing $M(\mathcal{C})$, the planar map induced by $\mathcal{C}$. Every connected component of $M(\mathcal{C})$ can be oriented independently, therefore we describe the algorithm assuming $M(\mathcal{C})$ is connected. Suppose $\mathcal{C}$ contains a curve $c$ which is crossed less than 3 times. By removing $c$ we obtain a set of $n-1$ curves in which there must be at least two curves (the ones crossed by $c$) which are crossed at most twice. We continue removing the curves, until none is left. Then we reinsert the curves in a reverse order (the last to be removed will be the first to be reinserted and so on). During the insertion process we reconstruct $M(\mathcal{C})$ and define a total order of its vertices. For this purpose we store the vertices of $M(\mathcal{C})$ in a data structure suggested by Dietz and Sleator [1]. This data structure supports the following operations, both in $O(1)$ worst-case time:

1. `INSERT`($X$,$Y$): Insert a new element $Y$ immediately after the element $X$.
2. `ORDER`($X$,$Y$): Compare $X$ and $Y$.

Note that by inserting $Y$ after $X$ and then switching their labels we can also use this data structure to insert a new element immediately *before* an existing element in constant time. We also keep a record of the maximal element in the order, MAX (that is, we update MAX when a new element is added after it).

We now describe the way a curve $c$ is reinserted. For every curve $c'$ that has already been added and is crossed by $c$ (recall that there are at most two such curves) we take the following actions. Let $x$ be the crossing point of $c$ and $c'$. If $c'$ has no other crossing points, then $x$ is inserted after MAX. In case $c'$ has exactly one crossing point $x'$, we insert $x$ after $x'$ when $c'$ is oriented from $x'$ to $x$, and before $x'$ otherwise. Otherwise, suppose $c'$ has two crossing points $x'_1$ and $x'_2$, such that $x'_1 < x'_2$. Then we insert $x$ before $x'_1$ if $x'_1$ is the middle point on $c'$ among the three points; after $x'_1$ if $x$ is the middle point; and after $x'_2$ if $x'_2$ is the middle point. Finally we orient $c$ arbitrarily if it has less than two crossings, or from the smaller crossing to the greater one, in case it has two crossings. We refer to the algorithm described above as Algorithm 1.

**Lemma 3** *Let $\mathcal{C}$ be a set of $n$ curves such that every curve is crossed at most three times and there is a curve that is crossed at most twice. Then the Algorithm 1 finds an acyclic orientation of $\mathcal{C}$ in $O(n)$ time.*

The more complicated case is when all the curves in $\mathcal{C}$ are crossed exactly three times. However, in this short version of the paper we only provide the general idea in this case, which is to:

1. find a set of curves $S$ that form a 'special' type of undirected cycle in $M(\mathcal{C})$;
2. orient $\mathcal{C} \setminus S$ using Algorithm 1; and
3. orient $S$ such that:
   (a) the curves in $S$ do not form a directed cycle; and
   (b) it is impossible to 'hop' on $S$ from $\mathcal{C} \setminus S$, 'travel' on $S$, and 'hop' off back to $\mathcal{C} \setminus S$.

## 2.2 Curves with at most five crossings per curve

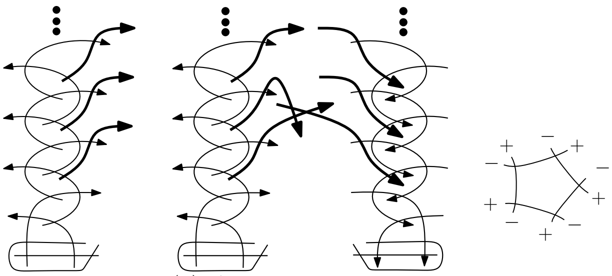
In the section we show that deciding whether there exists an acyclic orientation of a set of curves with at most 5 crossings per curve is intractable. We will reduce this problem from the Not-All-Equal-$k$-SAT[1] ($k \geq 3$) problem which is known to be NP-complete [7].

**Proof of Theorem 2.** An acyclic orientation can be verified in polynomial time, therefore the problem is in NP. The problem is shown to be NP-hard by reduction from Not-All-Equal-$k$-SAT to the acyclic orientation problem for $k \geq 3$.

Note that drawing the NAE problem in the plane introduces crossings between the wires. We call these crossings *extra*-crossings in order to distinguish them from the crossings between the curves. A variable is encoded as shown in Figure 2(a) where orientations correspond to Boolean signals. In any acyclic orientation all the curves drawn as arrows either have the orientation depicted or the opposite. The thick curves are used as wires and can have three further crossings. The construction uses $4+3k$ curves to generate $k$ wires.

Figure 2(b) shows the encoding of a NOT gate. It uses two wires from one variable and one from the other. The latter is used to propagate the signal across an extra-crossing without introducing a sixth crossing on a curve (note that this wire has only four

[1] Not-All-Equal-$k$-SAT ($k \geq 3$) is given by a collection of clauses, each containing exactly $k$ literals. The problem is to determine whether there exists a truth assignment such that each clause has at least one true and one false literal.



(a) A variable (b) A gadget for handling negation and extra-crossings (c) A clause

Figure 2: A reduction from NAE-$k$-SAT to orientation of curves with at most 5 crossings per curve.

crossings). Using this gadget a signal is negated before and after the extra-crossing, thus preventing the introduction of new cycles through the extra-crossing point.

The encoding of a clause with $k$ literals is done by $k$ curves forming a $k$-gon, as shown in Figure 2(c) for $k = 5$. A wire enters at the plus or at the minus sign depending on whether its corresponding literal in the clause in negated. The edges of the $k$-gon form a directed cycle if and only if all the literals of the clause are true or all are false. A solution to the Not-All-Equal-$k$-SAT problem will therefore yield an acyclic orientation of the curves. Conversely, if there is no solution, any orientation will either have a cycle at a clause encoding, or have outgoing edges at a variable encoding with different orientations, forcing a cycle within the variable. □

## 3 Acyclic orientation of topological graphs

Given a topological graph in which no edge is crossed, one can use the simple algorithm for abstract graphs described in the Introduction to find an acyclic orientation. Thus, the first non-trivial case is when every edge is crossed as most once.

### 3.1 Topological graphs with at most one crossing per edge

**Theorem 4** *Let $G$ be a simple topological graph on $n$ vertices in which every edge is crossed at most once. Then $G$ has an acyclic orientation. Moreover, such an orientation can be found in $O(n)$ time.*

Before proving Theorem 4 we recall a basic term from graph theory [3].

**Definition 1** *Given a biconnected graph $G = (V, E)$ and an edge $(s, t) \in E$, an* st-numbering *(or* st-ordering*) of $G$ is a bijection $\ell : V \to \{1, 2, \ldots, |V|\}$ such that: (a) $\ell(s) = 1$; (b) $\ell(t) = |V|$; and (c) for every vertex $v \in V \setminus \{s, t\}$ there are two edges $(v, u), (v, w) \in E$ such that $\ell(v) < \ell(u)$ and $\ell(v) > \ell(w)$.*

Given an st-numbering we will not make a distinction between a vertex and its st-number. An st-ordering of a graph $G$ naturally defines an orientation of the edges of $G$: direct every edge $(u, v)$ from $u$ to $v$ if $u < v$ and from $v$ to $u$ otherwise. The proof of the next lemma is omitted due to lack of space.

**Lemma 5** *Let $G = (V, E)$ be a plane biconnected multi-graph such that $|V| > 2$, and denote by $G'$ the directed planar graph defined by some st-numbering of $G$. Let $f$ be a face of $G$ (and $G'$), and denote by $G'_f$ (resp., $G_f$) the graph induced by the edges of $G'$ (resp., $G$) bounding $f$. Then $G'_f$ has exactly one source and one sink.*

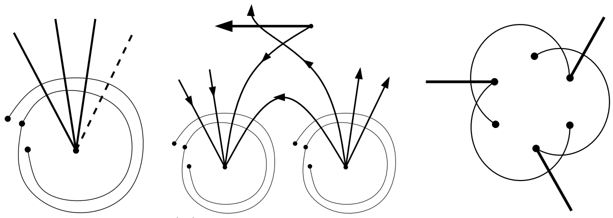
**Algorithm 3** Acyclic orientation of a topological graph with at most one crossing per edge

**Input**: A topological graph $G$ with at most one crossing per edge.
**Output**: An acyclic orientation of $G$.

1: **for** each pair of crossing edges $(a, b)$ and $(c, d)$ **do**
2: add the edges $(a, c), (a, d), (b, c)$, and $(b, d)$;
3: **end for**
4: compute the biconnected components of the new graph;
5: **for** each biconnected component $C$ **do**
6: delete all pairs of crossing edges in $C$;
7: find an st-numbering of the remaining graph;
8: reinsert all pairs of crossing edges in $C$;
9: orient each edge of $C$ according to the st-numbering;
10: **end for**
11: remove the edges added in line 2;

**Proof of Theorem 4.** Let $G$ be a simple topological graph in which every edge is crossed at most once. Denote by $n$ the number of vertices in $G$, and by $m$ the number of its edges. We will show that Algorithm 3 computes an acyclic orientation of $G$. Denote by $G'$ the graph obtained after adding the edges in lines 1–3. Note that it is always possible to add the edges listed in line 2 without introducing new crossings. After this step the vertices of each crossing pair of edges lie on a simple 4-cycle. It is enough to verify that each biconnected component of $G'$ is acyclicly oriented, since (a) every simple cycle in the underlying abstract graph is contained entirely in some biconnected component; and (b) the crossings do not introduce any interaction between different biconnected components, as all the vertices of a crossing pair of edges lie on a simple 4-cycle and therefore are in the same biconnected component. Thus, for the rest of the proof we assume $G'$ is biconnected. We denote by $G''$ the graph obtained by removing all the pairs of crossing edges. Removing a chord from a cycle does not affect the connected components of a graph, thus $G''$ is biconnected. Therefore, in line (7) an st-numbering of $G''$ is indeed computed.

Clearly, one can obtain an acyclic orientation of an abstract graph by numbering the vertices of the graph and directing every edge from its endpoint with the smaller number to its endpoint with the larger number. Therefore, it is enough to verify that the crossing



(a) A variable (b) A negation gadget with an edge (wire) crossed only once so far (c) A clause gate

Figure 3: A reduction from NAE-$k$-SAT to acyclic orientation of a topological graph with at most 4 crossings per edge.

points do not introduce a *bad* "shortcut", that is a path from a vertex $u$ to a vertex $v$ such that $v < u$. Let $((a, b), (c, d))$ be a pair of crossing edges. Denote by $f$ the 4-face $a - c - b - d - a$ of $G''$. According to Lemma 5 the digraph induced by $f$ and the computed st-numbering has only one source and sink. Therefore, we have to consider only two cases based on whether the sink and the source are adjacent in $f$. One can easily verify by inspection that in both cases no bad shortcut is formed. Thus Algorithm 3 produces an acyclic orientation.

Note that Algorithm 3 can be implemented to run in time linear in the number of vertices: Finding the biconnected components of a graph takes $O(n + m)$ time [2], as does the computation of an st-numbering [3]. Therefore the overall running time is $O(n + m)$, however the maximum number of edges in a topological graph in which every edge is crossed at most once is $4n - 8$ [6]. □

### 3.2 Topological graphs with at most four crossings per edge

**Theorem 6** *Let $G$ be a simple topological graph on $n$ vertices in which every edge is crossed at most once, and each curve has at most four crossings. Then deciding whether $G$ has an acyclic orientation is NP-complete.*

**Proof** (sketch). As for the case of a set of curves, the reduction is done from NOT-ALL-EQUAL-$k$-SAT. The gadgets we use appear in Figure 3. □

## 4 Discussion

We considered the problem of finding an acyclic orientation for a given topological graph or a set of curves in the plane. For topological graphs with at most one crossing per edge we showed an algorithm for finding an acyclic orientation in linear time. It follows from our results that when the maximum crossing per edge is at least four, deciding whether an acyclic orientation of the graph exists is NP-complete. An obvious open question is what happens when the maximum number of crossings per edge is two or three. A non-orientable topological graph with at most three crossings per edge can be constructed by combining the gadgets shown in Figure 3(a) (without the dashed wire) and in Figure 3(c). However, deciding whether a topological graph with at most three crossings per edge has an acyclic orientation is open. The situation is worse for topological graphs with at most two crossings per edge: So far we were unable to find an example which has no acyclic orientation, or to prove that every such graph is acyclic-orientable.

A special case is where all the vertices in the topological graph have degree 1. This case corresponds to asking the acyclic orientation question for a set of curves. Clearly, if the problem can be solved (or decided) for topological graphs with at most $k$ crossings per edge, then it can be solved for curves with at most $k$ crossings per curve. It would be interesting to determine whether there is a construction that provides a reduction from topological graphs with at most $k$ crossings per edge to a set of curves with at most $k'$ crossings per curve.

For curves with at most three crossings per curve we provided a linear time algorithm that finds an acyclic orientation. For five crossings per curve we showed that the problem becomes NP-complete. A set of curves in which every curve is crossed at most four times might not have an acyclic orientation, as Figure 1 implies. However, the decision problem for such sets of curves is also open. Two other interesting open questions are: (1) What happens if we only require acyclic *faces*? and (2) What happens if we look for an orientation such that for every pair of vertices, $u, v$, in the induced planar map there is a directed path from $u$ to $v$ or vice versa?

**Acknowledgements** We thank Michel Pocchiola for suggesting a problem that led us to study the questions discussed in this paper. We also thank Scot Drysdale, Frank Hoffmann, and Klaus Kriegel for helpful discussions.

## References

[1] P. DIETZ AND D. SLEATOR, Two algorithms for maintaining order in a list, *Proc. 19th Ann. ACM Symp. on Theory of Computing* (STOC), NYC, NY, 1987, 365–372.

[2] S. EVEN, *Graph Algorithms*, Computer Science Press, 1979.

[3] S. EVEN AND R. E. TARJAN, Computing an st-numbering, *Theoretical Computer Science*, 2(3):339–344, 1976.

[4] A. GRIGORIEV AND H. L. BODLAENDER, Algorithms for graphs embeddable with few crossings per edge, *Proc. 15th Int. Symp. on Fundamentals of Computation Theory* (FCT), Lübeck, Germany, *Lecture Notes in Computer Science*, volume 3623, Springer, 378–387, Sep. 2005.

[5] J. PACH, R. RADOICIC, G. TARDOS, AND G. TÓTH, Improving the crossing lemma by finding more crossings in sparse graphs, *Proc. 20th ACM Symp. on Computational Geometry* (SoCG), Brooklyn, NY, 2004, 68–75.

[6] J. PACH AND G. TÓTH, Graphs drawn with few crossings per edge, *Combinatorica*, 17(3):427–439, 1997.

[7] T. J. SCHAEFER, The complexity of satisfiability problems, *Proc. 10th Ann. ACM Symp. on Theory of Computing* (STOC), San Diego, CA, 1978, 216–226.