A Parameterized Algorithm for Upward Planarity Testing
(Extended Abstract)

Hubert Chan*

School of Computer Science
University of Waterloo, Canada
hy3chan@uwaterloo.ca

Abstract. Upward planarity testing, or checking whether a directed graph has a drawing in which no edges cross and all edges point upward, is NP-complete. All of the algorithms for upward planarity testing developed previously focused on special classes of graphs. In this paper we develop a parameterized algorithm for upward planarity testing that can be applied to all graphs and runs in $O(f(k)n^3 + g(k,\ell)n)$ time, where n is the number of vertices, k is the number of triconnected components, and ℓ is the number of cutvertices. The functions $f(k)$ and $g(k,\ell)$ are defined as $f(k) = k!8^k$ and $g(k,\ell) = 2^{3\cdot 2^\ell} k^{3\cdot 2^\ell} k!8^k$. Thus if the number of triconnected components and the number of cutvertices are small, the problem can be solved relatively quickly, even for a large number of vertices. This is the first parameterized algorithm for upward planarity testing.

1 Introduction

The area of graph drawing deals with geometric representations of abstract graphs, and has applications in many different areas such as software architecture, database design, project management, electronic circuits, and genealogy. These geometrical representations, known as graph drawings, represent each vertex as a point on the plane, and each edge as a curve connecting its two endpoints. Broader treatments of graph drawing are given by Di Battista et al. [6] and by Kaufmann and Wagner [21]. In our discussion of algorithmic results for graphs, we will use the number of vertices, n, as the input size.

Upward planarity testing, or determining whether or not a directed graph can be drawn with no edge crossings such that all edges are drawn upward, is NP-complete [17]. An example of an upward planar drawing is shown in Figure 1. In this paper, we will show how to apply parameterized complexity techniques to solve upward planarity testing. We show how to test upward planarity in $O(k!8^k n^3 + 2^{3\cdot 2^\ell} k^{3\cdot 2^\ell} k!8^k n)$ time, where k is the number of triconnected components in the graph and ℓ is the number of cutvertices. If the graph is biconnected, we give a $O(k!8^k n^3)$ time algorithm. Thus if k is small, we can do upward planarity testing efficiently.

* Research supported by an NSERC Postgraduate Scholarship.

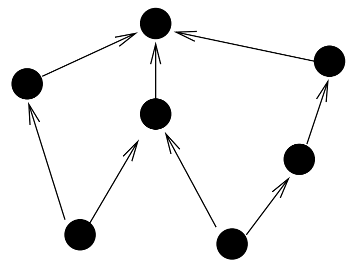

Fig. 1. An example of an upward planar drawing

A common approach to solve the problem of upward planarity testing is to look at special classes of graphs. Polynomial-time algorithms have been given for *st*-graphs [9], bipartite graphs [8], triconnected graphs [2], outerplanar graphs [22], and single-source graphs [3, 19]. Di Battista and Liotta [7] give a linear time algorithm that checks whether a given drawing is upward planar. Combinatorial characterizations for upward planarity are given by Tamassia and Tollis [25], Di Battista and Tamassia [9], Thomassen [27], and Hutton and Lubiw [19].

For some applications, upward planarity requirements may not be sufficient, or may be too strict. Jünger et al. [20] investigate proper layered drawings, Bertolazzi et al. [1] introduce quasi-upward planarity, and Di Battista et al. [10] investigate the area requirements for upward drawings.

Parameterized complexity, introduced by Downey and Fellows [12] is a technique to develop algorithms that are polynomial in the size of the input, but possibly exponential in one or more parameters, and hence efficient for small fixed parameter values. Although this is the first application of parameterized complexity to upward planarity testing, it has been applied to solve some other problems in graph drawing. Peng [23] investigates applying the concept of treewidth and pathwidth to graph drawing. Parameterized algorithms have been developed for layered graph drawings [13, 14], three-dimensional drawings of graphs of bounded path-width [16], the one-sided crossing maximization problem [15], and the two-layer planarization problem [24].

This paper is organized as follows: in Section 2, we define terms that we use in this paper. In Section 3, we develop a parameterized algorithm for upward planarity testing in biconnected graphs. We will then show how to join together biconnected components: given two components G_1 and G_2 to joined at vertices $v_1 \in V(G_1)$ and $v_2 \in V(G_2)$, we draw one component within a face of the other, draw an edge from v_1 to v_2, and contract the edge. Thus we must investigate the effects of edge contraction in upward planar graphs (Section 4). We also define a notion of node accessibility in Section 5, which will determine when the edge (v_1, v_2) can be drawn. In Section 6, we use the results from the previous sections to join together biconnected components, giving us a parameterized algorithm for upward planarity testing in general graphs.

2 Definitions

In this paper, we assume the reader is familiar with the standard definitions of graphs and directed graphs, which can be found in a basic graph theory book [4, 11]. Given a graph G, we denote by $V(G)$ and $E(G)$ the sets of its vertices and edges, respectively. If there is no confusion as to which graphs we refer, we may simply write V and E. Unless otherwise specified, all graphs in this paper are directed simple graphs. In this paper, we adopt the convention of using lowercase Roman letters to represent vertices and lowercase Greek letters to represent edges.

A graph is *biconnected* (*triconnected*) if between any two vertices, there are at least two (three) vertex-disjoint paths. A *cutvertex* is a vertex whose removal disconnects the graph.

Given vertex v in a directed graph, its *indegree* (*outdegree*), denoted $\deg^-(v)$ ($\deg^+(v)$), is the number of incoming (outgoing) edges. A vertex with no incoming (outgoing) edges is called a *source* (*sink*).

A *drawing* φ of a graph G is a mapping of vertices to points on the plane, and edges to curves such that the endpoints of the curve are located at the vertices incident to the edge. In the case of directed graphs, each curve has an associated direction corresponding to the direction of the edge. We say that a curve is *monotone* if every horizontal line intersects the curve at most once. A graph is *planar* if it has a drawing in which no edges cross. A directed graph is *upward planar* if it has a planar drawing in which all edges are monotone and directed upwards.

A planar drawing partitions the plane into regions called *faces*. The infinite, or unbounded, face is called the *outer face*.

A drawing φ defines, for each vertex v, a clockwise ordering of the edges around v. The collection Γ of the clockwise orderings of the edges for each vertex is called a *(planar) embedding*. If edge ϵ_2 comes immediately after ϵ_1 in the clockwise ordering around v, then we say that ϵ_1 and ϵ_2 are *edge-ordering neighbours*, ϵ_2 is the *clockwise neighbour* of ϵ_1 around v, and ϵ_1 is the *counter-clockwise neighbour* of ϵ_2.

In an upward planar drawing, all incoming edges to v must be consecutive, as must all outgoing edges [25]. Thus for each vertex, we can partition the clockwise ordering of its edges into two linear lists of outgoing and incoming edges. We call the collection of these lists an *upward planar embedding*. If v is a source (sink), then the first and last edges in the list of outgoing (incoming) edges specify a face. Using the terminology of Bertolazzi et al. [2], we say that v is a *big angle* on that face.

If a vertex v has incoming edge ϵ_1 and outgoing edge ϵ_2 that are edge ordering neighbours, then we say that the face that contains ϵ_1 and ϵ_2 as part of its boundary is *flattenable* at v.

We *contract* an edge $\epsilon = (v, w)$ by removing v and w, along with their incident edges. We then add a new vertex v_ϵ, and for each edge (u, v) or (u, w) in $E(G)$, we add the edge (u, v_ϵ). For each edge (v, u) or (w, u), we add the edge (v_ϵ, u). The resulting graph is denoted G/ϵ. Given an embedding Γ of G, we construct an embedding of G/ϵ, denoted Γ/ϵ, as follows: for each vertex $u \neq v, w$, the clockwise ordering of edges around u remains the same. We then construct the clockwise ordering around v_ϵ by first adding the edges incident to v in order, starting with the clockwise neighbour of ϵ around v and ending with the counter-clockwise neighbour of ϵ, then adding the edges incident to w in order, starting with the clockwise neighbour of ϵ around w and ending with the counter-clockwise neighbour of ϵ. We note that Γ/ϵ may not be an upward planar embedding, even if Γ was; we investigate this further in Section 4.

3 Biconnected components

We first consider only biconnected components; in Section 6, we show how to connect these components together. Our goal for this section is to bound the number of possible planar embeddings of a biconnected graph by a function $f(k)$, where k is the number of triconnected components in the graph, which will allow us to obtain a parameterized algorithm for upward planarity testing.

Theorem 1. *Given a planar biconnected graph G that has k triconnected components, G has at most $k!8^{k-1}$ possible planar embeddings, up to reversal of all the edge orderings.*

Proof. (outline) We first show that given two vertices in an embedded triconnected graph, there are at most two faces that contain both vertices. From this, we can show that given two triconnected components G_1 and G_2 of G that share a common vertex v, there are at most eight possible embeddings of $G_1 \cup G_2$. The eight embeddings come from the fact that there are at most two faces of G_1 in which G_2 can be drawn (which are the two faces that contain both v and another vertex that lies on a path from G_1 to G_2), and vice versa, and that there are two possibilities for the edge orderings of G_2 with respect to G_1. From this, we can show that if G has k triconnected components $G_1, \ldots, G_k$ that all share a common vertex, there are at most $(k-1)!8^{k-1}$ possible embeddings of $G_1 \cup \cdots \cup G_k$. Since there are at most k shared vertices between triconnected components, we have at most $k(k-1)!8^{k-1} = k!8^{k-1}$ embeddings. □

Using this bound, we can produce a parameterized algorithm that tests whether G is upward planar.

Theorem 2. *There is an $O(k!8^k n^3)$-time algorithm to test whether a biconnected graph is upward planar, where n is the number of vertices and k is the number of triconnected components.*

Proof. Our algorithm works as follows: first it divides the input graph G into triconnected components, which can be done in quadratic time [18]. It then tests each possible embedding of G for upward planarity. By Theorem 1, we have $k!8^{k-1}$ embeddings. From Euler's formula, we know that for each embedding, there are at most n possible outer faces. Bertolazzi et al. [2] give a quadratic-time algorithm to determine whether a given embedding and outer face correspond to an upward planar drawing. Thus we can run the algorithm for each possible embedding and outer face, giving a time complexity of $O(k!8^k n^3)$. □

Our bound on the number of possible embeddings is, in many cases, much larger than the actual number of possible embeddings. For example, if each vertex is common to at most two triconnected components, we have only 8^{k-1} possible embeddings, rather than $k!8^{k-1}$. We also note that Theorem 1 does not depend on upward planarity. Therefore a similar technique could be applied to other graph drawing problems in which we have an algorithm that solves the problem given a specific embedding.

4 Edge contraction

We now investigate how we can join upward planar embeddings of biconnected graphs in order to obtain an upward planar embedding of a general graph. The joining is achieved by first connecting the biconnected graphs with an edge, and then contracting the edge. Thus we will first examine the effect of edge contraction on upward embeddings. Notably, we will determine conditions under which contraction of an edge in an upward planar embedding produces an embedding that is still upward planar.

Throughout this section, we let G be an upward planar graph with upward planar embedding Γ, and let $\epsilon = (s,t)$ be the edge that we wish to contract. We also assume that both s and t have degree greater than one; if not, it is easy to see that contracting ϵ will result in an upward planar embedding. Thus we will consider the edge-ordering neighbours of ϵ. Throughout this section, we will use the following labels. Let $\alpha = (a,t)$ and $\beta = (b,t)$ be the clockwise and counterclockwise neighbours, respectively, of ϵ around t in the embedding Γ, and let $\gamma = (c,s)$ and $\delta = (d,s)$ be the counterclockwise and clockwise neighbours of ϵ around s (Figure 2 a).

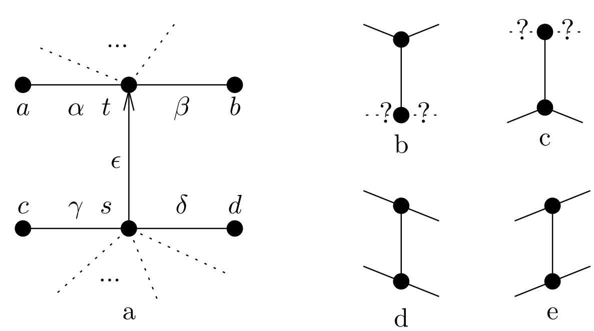

Fig. 2. The vertices around the edge ϵ, and the edge orientations that we consider.

Since ϵ is an arbitrary edge, we must consider the orientations of α, β, γ, and δ. As shorthand, if α or β is oriented towards t, we say that it is *oriented inward*, and similarly for when γ or δ is oriented towards s. If α or β is oriented away from t, we say that it is *oriented outward*, and similarly for γ and δ.

In this paper, we will only consider the four cases for the orientations of α, β, γ, and δ that will be used in Section 6, and we only give the lemma statements. The remaining cases, as well as the complete proofs, are given in the thesis from which this work is derived [5].

Case 1 α and β oriented outward, γ and δ oriented arbitrarily (Figure 2b),
Case 2 γ and δ oriented inward, α and β oriented arbitrarily (Figure 2c),
Case 3 α and γ oriented outward, β and δ oriented inward (Figure 2d), and
Case 4 α and γ oriented inward, β and δ oriented outward (Figure 2e)

In all four cases, we show that G/ϵ is upward planar with embedding Γ/ϵ. The proof of Lemma 1, which proves Cases 1 and 2, is a straightforward extension of a lemma by Hutton and Lubiw [19], and Lemma 2, which proves Cases 3 and 4, can be easily shown using the characterization given by Bertolazzi et al. [2]. We omit both proofs.

Lemma 1. *If* $\deg^-(t) = 1$ *(*$\deg^+(s) = 1$*), then* G/ϵ *is upward planar with upward planar embedding* Γ/ϵ*.* □

Lemma 2. *If the edges* α *and* γ *are oriented outward (inward), and* β *and* δ *are oriented inward (outward), then* G/ϵ *is upward planar with upward planar embedding* Γ/ϵ*.* □

5 Node Accessibility

We now define a notion of accessibility of vertices from different parts of the outer face, namely the area above or below the drawing of G. This, along with the edge contraction results from the previous section, will help us join together upward planar subgraphs.

Given an upward planar graph G with a specified upward planar embedding Γ and outer face F, we say that the vertex v is *accessible from above (below)* if there is an upward planar drawing of G corresponding to the specified embedding and outer face such that a monotone curve that does not cross any edges can be drawn from v to a point above (below) the drawing of G.

Note that if a monotone curve can be drawn from v to a point p above the drawing of G, we can draw a monotone curve from v to any other point q above the drawing of G by appropriately modifying the curve from v to p. Therefore our definition of accessibility from above is not dependent on the point to which we draw the curve.

In order to efficiently test whether or not a vertex is accessible from above or from below, we wish to derive conditions for vertex accessibility that we can obtain directly from the planar embedding, rather than from a graph drawing. With the conditions below, we can test whether a vertex v is accessible from above or from below in $O(n)$ time.

Theorem 3. *If the vertex v is on the outer face and has an outgoing (incoming) edge ϵ that is on the outer face, then v is accessible from above (below).*

Proof. (outline) We can first show an equivalent definition of accessibility from above and from below: v is accessible from above if there is an upward planar drawing of G such that we can draw a monotone curve from v to a point p that is above v and on the outer face. The point p need not be above the drawing of G. Proving that this definition is equivalent is long and largely technical. To prove it, we show how to take a drawing in which we can draw a monotone curve from v to p, and modify this drawing so that we can draw a new curve from v to a new point q that is above the drawing of G.

It is then easy to see that if v has an outgoing edge on the outer face, then we can draw a curve from v to a point above v on the outer face. □

Theorem 4. *A source (sink) v is accessible from below (above) if and only if it is a big angle on the outer face.*

Proof. As shown by Bertolazzi et al. [2], in any upward planar drawing of G, if v is a big angle on the outer face, then the angle of the outer face formed at v must be greater than 180°, and hence the outer face must include some area below v. Thus using the alternate definition of accessibility from the proof of the previous theorem, we can draw a monotone curve from v to a point below v on the outer face, and hence v is accessible from below.

To show the other direction, we note that if v is a source, there is only one face below v, and hence v must be a big angle on that face. Since v is accessible from below, that face must be the outer face. □

6 Joining biconnected components

We are now ready to show how to join multiple upward planar biconnected components to form a larger upward planar graph. Throughout this section, we let $G_1, \ldots, G_k$, with upward planar embeddings $\Gamma_1, \ldots, \Gamma_k$ and outer faces $F_1, \ldots, F_k$, be connected (not necessarily biconnected) components. We will start by joining the components by identifying the vertices $v_1 \in V(G_1), \ldots, v_k \in V(G_k)$. The resulting graph we call G, with upward planar embedding Γ. By repeating this process, we can construct any arbitrary graph, starting from its biconnected components.

To help us prove our conditions for joining graphs, we first show that for all i, except for at most one, v_i must be on the outer face of the corresponding G_i in order for G to be upward planar.

Lemma 3. *If there exist more than one value of i such that G_i does not have an upward planar embedding with v_i on the outer face then G is not upward planar.*

Proof. (outline) We assume that we are given an upward planar drawing of G, and show that if we consider this drawing, restricted to the subgraph G_i, then v_i must be on the outer face for $i > 1$. □

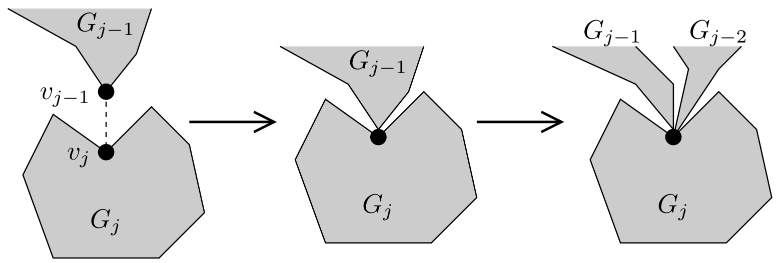

Fig. 3. Joining G_j to G_{j-1} and then to G_{j-2} when v_j, v_{j-1}, and v_{j-2} are all sources.

We identify three cases that we will use to join together components to construct arbitrary graphs.

Case 1 All v_i are sources, or are all sinks.
Case 2 $k = 2$ and v_1 is a source or a sink.
Case 3 None of the v_i's are sources or sinks, and are not cutvertices.

The way in which we will use these cases is as follows: we will join together all components in which v_i is a source to form a larger component by using Case 1. Similarly, we join together all components in which v_i is a sink. Using Case 3, we join together all components in which v_i is neither a source nor a sink. Finally, using Case 2, we join together the three new components. In each case, we give necessary and sufficient conditions for the upward planarity of G.

Theorem 5. *If v_i is a source (sink) for all i, then G is upward planar if and only if for all $j > 1$, G_j has an upward planar drawing in which v_j is on the outer face.*

Proof. Since v_j is on the outer face and is a source, it must have an outgoing edge on the outer face, and so by Theorem 3, must be accessible from above. So we can draw G_{j-1} in the outer face of G_j above v_j, draw an edge from v_j to v_{j-1} and contract the edge, using Lemma 1, as illustrated in Figure 3. We start this procedure from the last component, and note that if v_j and v_{j-1} are on the outer faces in their respective components, then after v_j and v_{j-1} have been identified, the new vertex is still a source on the outer face of the new component.

The other direction follows directly from Lemma 3. □

Theorem 6. *Given two components G_1 and G_2 where v_1 is a source (sink), G is upward planar if and only if G_1 has an upward planar embedding in which v_1 is accessible from below (above), or G_2 has an upward planar embedding in which v_2 is accessible from above (below).*

Proof. (outline) We omit the proof that the condition is sufficient as the proof is similar to the proof above. To show the other direction, we assume that G

is upward planar, v_1 is not accessible from below, and v_2 is not accessible from above. We first show that if v_1 and v_2 are not cutvertices of their respective components, then G is not upward planar. We do this by showing that if v_1 and v_2 are cutvertices, then G_1 is drawn entirely within a single face of G_2. We then consider the faces of G_2 in which G_1 can be drawn, and show that in every case we obtain a contradiction.

We then show that if v_1 is a cutvertex, we can find a subgraph of G_1 in which v_1 is not a cutvertex and is not accessible from below. Similarly, if v_2 is a cutvertex, we can find a subgraph of G_2 in which v_2 is not a cutvertex and is not accessible from above. From the result above, these two subgraphs cannot be joined to yield an upward planar graph. Since this subgraph of G is not upward planar, G cannot be upward planar. □

Theorem 7. *If for all i v_i has indegree and outdegree at least 1 and v_i is not a cutvertex, then G is upward planar if and only if for all $i > 1$, G_i has an upward planar embedding in which the outer face F_i is flattenable at v_i.*

Proof. (outline) Again, we omit the proof that the condition is sufficient. We assume that we are only given two components G_1 and G_2 such that F_i is not flattenable at v_i: if we have more than two components such that F_i is not flattenable at v_i for more than one value of i, we can take any two such components as G_1 and G_2 and show that G is not upward planar. Again, since v_1 and v_2 are not cutvertices, in any upward planar drawing of G, G_1 must be drawn entirely within a single face of G_2 and vice versa, and in all cases, we obtain a contradiction and hence G cannot be upward planar. □

We can now give a fixed-parameter algorithm for determining when a graph is upward planar, with the parameter being the number of triconnected components and the number of cutvertices.

Theorem 8. *There is an $O(k!8^k n^3 + 2^{3\cdot 2^\ell} k^{3\cdot 2^\ell} k!8^k n)$-time algorithm to test whether a graph G is upward planar, where n is the number of vertices, k is the number of triconnected components, and ℓ is the number of cutvertices.*

Proof. (outline) Our algorithm works by splitting G into biconnected components, generating all possible upward planar embeddings for the biconnected components, and joining them together, keeping track of all possible upward planar embeddings for the new components.

We can G split into biconnected components in $O(n)$ time [26]. For each biconnected component, we generate all possible upward planar embeddings, along with their possible outer faces, as shown in Section 3. In total, this takes $O(k!8^k n^3)$ time, and generates at most $k!8^{k-1}$ embeddings.

For each cutvertex v, we first join together all components in which v has indegree and outdegree at least one by using Theorem 7, producing the new component $G_\times$. We then use Theorem 5 to join together all components in which v is a source, producing $G_\vee$, and all components in which v is a sink, producing $G_\wedge$. Then, using Theorem 6, we join $G_\vee$ to $G_\times$ and join $G_\wedge$ to the resulting component, producing a new component G_v. We remove all the components that contained v, replace them with G_v, and continue inductively. In this step, we may also detect that no possible upward planar embedding exists.

Since the conditions given by the theorems in this section only depend on the accessibility of vertices or on vertices being on the outer face, we can greatly limit the number of embeddings that we must consider. For example, if G_i is not on the outer face (and hence not accessible from above or from below) in the embeddings Γ_1 and Γ_2 of G_v and we later join G_v to a component that shares a vertex with G_i, Γ_1 can be used to create an upward planar embedding of the new graph if and only if Γ_2 can also be used. Thus we only need to consider either Γ_1 or Γ_2.

We can show that, for the ith cutvertex, the number of embeddings that we will produce will be less than $2^{2^i} k^{2^i} k! 8^k$, and producing them will take at most $O(2^{3\cdot 2^i} k^{3\cdot 2^i} k! 8^k n)$ time. Since we have ℓ cutvertices, summing over all the steps for joining biconnected components gives a time of at most $O(2^{3\cdot 2^\ell} k^{3\cdot 2^\ell} k! 8^k n)$.

Thus in total, the algorithm runs in $O(k! 8^k n^3 + 2^{3\cdot 2^\ell} k^{3\cdot 2^\ell} k! 8^k n)$ time. □

7 Conclusions and future work

In this paper, we first developed a parameterized algorithm for upward planarity testing of biconnected graphs. This algorithm runs in $O(k! 8^k n^3)$ time, where k is the number of triconnected components. We then showed conditions under which contracting an edge in an upward planar graph results in a new graph that is still upward planar, and we introduced a notion of vertex accessibility from above and below. Using these results, we then gave necessary and sufficient conditions for joining biconnected graphs to form a new upward planar graph. This allowed us to obtain a parameterized algorithm for upward planarity testing in general graphs. Our algorithm runs in $O(k! 8^k n^3 + 2^{3\cdot 2^\ell} k^{3\cdot 2^\ell} k! 8^k n)$ time, where n is the number of vertices, k is the number of triconnected components, and ℓ is the number of cutvertices.

Our running time analysis contains many potential overestimations. Better analysis may yield a smaller running time. It would also be interesting to implement the algorithm and see how well it performs in practice. In particular, since the complexity analysis includes bookkeeping of embeddings that are not upward planar, it is very likely that the running time in practice will be much smaller than that given in Theorem 8.

Another possible research direction in applying parameterized complexity to upward planarity testing is obtaining a parameterized algorithm that determines whether or not a graph has a drawing in which at most $\frac{1}{k}$ of the edges point downward. For $k = \infty$, this is simply upward planarity testing. For $k = 2$, this is trivial: take any drawing of the graph in which no edge is drawn horizontally. Either this drawing, or the drawing that results from flipping it vertically, has at least half the edges pointing upward. Thus it is possible that between these two extremes, we may be able to obtain a parameterized result.

Parameterized complexity is a fairly new area and seeks to find efficient solutions to hard problems. Many problems in graph drawing have been shown to be NP-complete, and so parameterized complexity may be able to offer solutions to many of these problems. Some possible parameters worth examining are the height, width, or area of the drawing, the maximum indegree or outdegree in a graph, the number of faces in a planar graph, or the number of sources or sinks.

Acknowledgements

I gratefully acknowledge the suggestions of the anonymous referees.

References

1. Paola Bertolazzi, Giuseppe Di Battista, and Walter Didimo. Quasi-upward planarity. *Algorithmica*, 32:474–506, 2002.
2. Paola Bertolazzi, Giuseppe Di Battista, Giuseppe Liotta, and Carlo Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12:476–497, 1994.
3. Paola Bertolazzi, Giuseppe Di Battista, Carlo Mannino, and Roberto Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM Journal on Computing*, 27(1):132–196, 1998.
4. John Adrian Bondy and U. S. R. Murty. *Graph Theory with Applications*. North Holland, New York, 1976.
5. Hubert Chan. A parameterized algorithm for upward planarity testing of biconnected graphs. Master's thesis, School of Computer Science, University of Waterloo, May 2003.
6. Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, New Jersey, 1999.
7. Giuseppe Di Battista and Giuseppe Liotta. Upward planarity checking: "Faces are more than polygons" (extended abstract). In Sue H. Whitesides, editor, *Proceedings of Graph Drawing 1998: 6th International Symposium*, volume 1547 of *Lecture Notes in Computer Science*, pages 72–86. Springer-Verlag, 1998.
8. Giuseppe Di Battista, Wei-Ping Liu, and Ivan Rival. Bipartite graphs, upward drawings, and planarity. *Information Processing Letters*, 36:317–322, 1990.
9. Giuseppe Di Battista and Roberto Tamassia. Algorithms for plane representations of acyclic digraphs. *Theoretical Computer Science*, 61:175–198, 1988.
10. Giuseppe Di Battista, Roberto Tamassia, and Ioannis G. Tollis. Area requirement and symmetry display of planar upward drawings. *Discrete and Computational Geometry*, 7:381–401, 1992.
11. Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, New York, 2nd edition, 2000.
12. Rod G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, New York, 1997.
13. Vida Dujmović, Michael R. Fellows, Michael T. Hallett, Matthew Kitching, Giuseppe Liotta, Catherine McCartin, Naomi Nishimura, Prabhakar Ragde, Frances Rosamond, Matthew Suderman, Sue Whitesides, and David R. Wood. A fixed-parameter approach to two-layer planarization. In *Proceedings of the 9th International Symposium on Graph Drawing*, volume 2265 of *Lecture Notes in Computer Science*, pages 1–15, 2001.

14. Vida Dujmović, Michael R. Fellows, Michael T. Hallett, Matthew Kitching, Giuseppe Liotta, Catherine McCartin, Naomi Nishimura, Prabhakar Ragde, Frances Rosamond, Matthew Suderman, Sue Whitesides, and David R. Wood. On the parameterized complexity of layered graph drawing. In *Proceedings, 9th Annual European Symposium on Algorithms*, pages 488–499, 2001.
15. Vida Dujmović, Henning Fernau, and Michael Kaufmann. Fixed parameter algorithms for ONE-SIDED CROSSING minimization revisited. In Giuseppe Liotta, editor, *Proceedings of Graph Drawing 2003: 11th International Symposium*, volume 2912 of *Lecture Notes in Computer Science*. Springer, 2004.
16. Vida Dujmović, Pat Morin, and David R. Wood. Path-width and three-dimensional straight-line grid drawings of graphs. In Stephen G. Kobourov and Michael T. Goodrich, editors, *Proceedings of Graph Drawing 2002*, volume 2528 of *Lecture Notes in Computer Science*, pages 42–53. Springer, 2002.
17. Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(2):601–625, 2001.
18. John Hopcroft and Robert E. Tarjan. Dividing a graph into triconnected compononts. *SIAM Journal on Computing*, 2:136–158, 1972.
19. Michael D. Hutton and Anna Lubiw. Upward planar drawing of single source acyclic digraphs. *SIAM Journal on Computing*, 25(2):291–311, 1996.
20. Michael Jünger, Sebastian Leipert, and Petra Mutzel. Level planarity testing in linear time. In S. Whitesides, editor, *Proceedings of Graph Drawing 1998: 6th International Symposium*, volume 1547 of *Lecture Notes in Computer Science*, pages 224–237. Springer, 1998.
21. Michael Kaufmann and Dorothea Wagner, editors. *Drawing Graphs: Methods and Models*, volume 2025 of *Lecture Notes in Computer Science Tutorial*. Springer, 2001.
22. Achilleas Papakostas. Upward planarity testing of outerplanar dags (extended abstract). In Roberto Tamassia and Ioannis G. Tollis, editors, *Proceedings of Graph Drawing 1994*, volume 894 of *Lecture Notes in Computer Science*, pages 298–306. Springer, 1995.
23. Zhou Peng. Drawing graphs of bounded treewidth/pathwidth. Master's thesis, Department of Computer Science, University of Auckland, 2001.
24. Matthew Suderman and Sue Whitesides. Experiments with the fixed-parameter approach for two-layer planarization. In Giuseppe Liotta, editor, *Proceedings of Graph Drawing 2003: 11th International Symposium*, volume 2912 of *Lecture Notes in Computer Science*. Springer, 2004.
25. Roberto Tamassia and Ioannis G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete and Computational Geometry*, 1(4):321–341, 1986.
26. Robert E. Tarjan. Depth-first searcth and linear graph algorithms. *SIAM Journal on Computing*, 1(2):145–159, 1972.
27. Carsten Thomassen. Planar acyclic oriented graphs. *Order*, 5:349–361, 1989.