

Upward Planarity Testing of Embedded Mixed Graphs*

Carla Binucci and Walter Didimo

Dip. Ing. Elettronica e dell'Informazione
Università degli Studi di Perugia
{binucci, didimo}@diei.unipg.it

Abstract. A *mixed graph* has both directed and undirected edges. We study an upward planarity testing problem for embedded mixed graphs and solve it using Integer Linear Programming. Experiments show the efficiency of our technique.

1 Introduction

An *upward planar drawing* of a planar digraph G is a planar drawing of G such that all the edges are drawn as curves monotonically increasing in the vertical direction, according to their orientation. The *upward planarity testing* problem is the problem of deciding whether a planar digraph admits an upward planar drawing, and has a long tradition in graph drawing [8]. It is polynomially solvable if the planar embedding of the graph is fixed [2], while it is NP-hard in the variable embedding setting [9].

Many graphs arising from real applications have both directed and undirected edges. These types of graphs are called *mixed graphs* and have received considerable attention in the literature (see, e.g., [1]). Fig. 1(a) shows a mixed graph whose nodes represent employees of a company; the directed edges describe hierarchical relationships while the undirected edges describe collaborations. In a visual representation of a mixed graph it is still desirable that directed edges flow upward, as in Fig. 1(b). Additionally, in order to increase the readability of the layout, one may want that even the undirected edges are drawn as curves vertically monotone when possible, as in Fig. 1(c). An *upward drawing* of a mixed graph G is such that all the directed edges of G are drawn upward and all the undirected edges of G are drawn monotone in the vertical direction.

In this paper we study the following problem: *Given an embedded planar mixed graph G , decide whether G admits an upward planar drawing that preserves the planar embedding of G .* The drawing in Fig. 1(c) is an embedding-preserving upward planar drawing of the graph in Fig. 1(a), while the drawing in Fig. 1(b) is not an upward drawing, because edge (Mary, Kate) is not vertically monotone. Our problem is equivalent to decide if there exists an orientation of the undirected edges of G such that the resulting embedded digraph has an upward planar drawing. The contribution of this paper is twofold:

- We describe an ILP (Integer Linear Programming) model for the upward planarity testing problem of a planar embedded mixed graph G ; the number of variables and

* Research supported in part by the MIUR project AlgoDEEP prot. 2008TFBWL4. We acknowledge Maurizio Patrignani for the useful discussion on the subject of this paper.

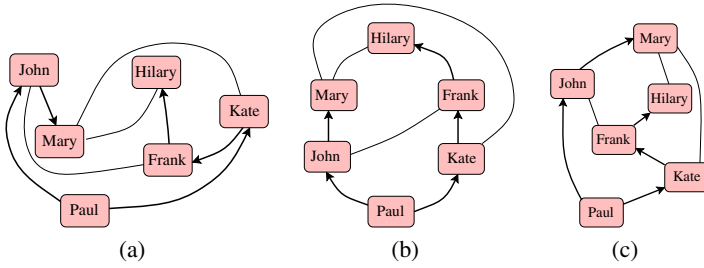


Fig. 1. (a) A planar embedded mixed graph G . (b) An embedding-preserving planar drawing of G , where the directed edges are drawn upward. (c) An embedding-preserving planar drawing of G where the directed edges are upward and the undirected edges are vertically monotone.

constraints of the model is linear in the size of G . If G has an embedding-preserving upward planar drawing, the model allows us to construct one (Section 3).

- We present an experimental study that shows how the proposed model can be solved efficiently (Section 4). Indeed, for all instances of our test suite the computation of a solution takes a few seconds, even for graphs with several hundreds of nodes.

We remark that the study of upward drawings of mixed graphs has been previously addressed by Eigsperger *et al.* [7]. Differently from our results, they describe a heuristic that attempts to compute an upward drawing with few edge crossings. Hence, they do not start from an embedded planar graph, and the final drawing may contain crossings even if the original graph admits an upward planar drawing according to our definition.

2 Definitions and Notation

Let G be an embedded planar digraph. A *source vertex* (resp. a *sink vertex*) of G is a vertex with only outgoing edges (resp. incoming edges). A source vertex or a sink vertex of G is also called a *switch vertex* of G . A vertex v of G is *bimodal* if all its incoming edges are consecutive around v (and thus also the outgoing edges are consecutive around v). If all vertices of G are bimodal then G and its embedding are called *bimodal*. Acyclicity and bimodality are necessary but not sufficient conditions for the upward planar drawability of an embedded planar digraph [2]. Note that, if G is bimodal, the circular list of edges incident to any vertex v of G is split into two linear lists, one consisting of the incoming edges of v and the other consisting of the outgoing edges of v .

Let f be a face of G and suppose that the boundary of f is visited clockwise if f is internal, and counterclockwise if f is external. Let $a = (e_1, v, e_2)$ be a triplet such that v is a vertex of the boundary of f and e_1, e_2 are two edges incident to v that are consecutive on the boundary of f (e_1 and e_2 may coincide if G is not biconnected). Triplet a is called an *angle at v in face f* , or simply an *angle of f* . An angle $a = (e_1, v, e_2)$ of a face f is a *switch angle* of f if e_1 and e_2 are both incoming edges or both outgoing edges of v ; otherwise a is a *non-switch angle*. If v is a switch vertex, all

the angles at v are switch angles in the faces incident to v . We denote by $\text{deg}(v)$ the number of angles at v and by $\text{deg}(f)$ the number of angles in f .

Let Γ be an upward planar drawing of an embedded planar digraph G . Assign to each angle a of G a label S , F , or L , according to the following rules: a is labeled L if it is a switch angle that corresponds to a geometric angle larger than π in Γ ; a is labeled F if it is a non-switch angle; a is labeled S otherwise. Note that, an angle is labeled S if it is a switch angle corresponding to a geometric angle smaller than π in Γ . We call this labeling the *upward labeling induced by Γ* . Given an embedded bimodal planar digraph G , an assignment \mathcal{L} of labels S , F , and L to the angles of G is called an *upward planar embedding of G* if there exists an upward planar drawing Γ of G such that the upward labeling induced by Γ coincides with \mathcal{L} . For a given angle labeling \mathcal{L} and for a given vertex v of G , we denote by $L(v)$, $S(v)$, and $F(v)$ the number of angles at v that are labeled L , S , and F , respectively; also, if f is a face of G , $L(f)$, $S(f)$, and $F(f)$ denote the number of angles of f that are labeled L , S , and F , respectively.

The next theorem characterizes the upward planar embeddings of an embedded bimodal planar digraph G . It is a consequence of the results in [2,5,6].

Theorem 1. *Let G be an embedded bimodal planar digraph and let \mathcal{L} be an assignment of labels S , F , and L to the angles of G . \mathcal{L} is an upward planar embedding of G if and only if the following properties hold: (a) $\text{deg}(f) - 2 = 2L(f) + F(f)$, for each internal face f of G ; (b) $\text{deg}(f) + 2 = 2L(f) + F(f)$, for the external face f of G ; (c) switch angles are labeled either S or L , and non-switch angles are labeled F ; (d) if v is a switch vertex of G then: $L(v) = 1$, $S(v) = \text{deg}(v) - 1$, $F(v) = 0$; (e) if v is a non-switch vertex of G then: $L(v) = 0$, $S(v) = \text{deg}(v) - 2$, $F(v) = 2$.*

Given an upward planar embedding \mathcal{L} of a digraph G , an upward planar drawing of G that induces \mathcal{L} can be computed efficiently by using algorithms described in [2,4].

3 An ILP Model

In order to decide whether an embedded planar mixed graph $G = (V, E)$ admits an upward planar drawing, we use the characterization of Theorem 1. Namely, we want to find an orientation for the undirected edges of G and a labeling \mathcal{L} for the angles of G such that the resulting digraph is bimodal and \mathcal{L} is an upward planar embedding of this embedded digraph. G' will denote the digraph obtained from G by orienting its undirected edges. To decide whether G' and \mathcal{L} exist we define an ILP model, whose sets, variables, and constraints are described below. We assume that the embedded digraph obtained from G by removing all the undirected edges is bimodal, otherwise we can immediately conclude that G does not have an upward planar drawing.

Sets. V is partitioned into two subsets V_{NS} and V_{PS} . Each vertex in V_{NS} has both incoming and outgoing edges, and therefore it cannot be a switch vertex of G' . Subset V_{PS} contains the remaining vertices of V ; each element in V_{PS} is a potential switch vertex of G' . E is partitioned into two subsets E_D and E_U , containing the directed and the undirected edges of G , respectively. F is the set of faces of G and $\text{adj}(v)$ is the set of neighbors of v .

A denotes the set of angles of G and is partitioned into subsets A_{NS} and A_{PS} , which contain the angles of G at vertices in V_{NS} and in V_{PS} , respectively. For a vertex v and for a face f , $A(v)$ and $A(f)$ denote all angles at v and all angles in f , respectively. For a vertex $v \in V_{NS}$, $A_{NS}(v)$ is the set of angles at v . For a vertex $v \in V_{PS}$, $A_{PS}(v)$ is the set of angles at v . If $v \in V_{NS}$, we denote by e'_{out} and e''_{out} the first and the last outgoing edge of v , respectively (e'_{out} and e''_{out} may coincide). Analogously, e'_{in} and e''_{in} are the first and the last incoming edges of v . The set of angles at v formed by the edges between e'_{in} and e'_{out} in clockwise order is denoted by $A^l_{NS}(v)$. The set of angles at v formed by the edges between e''_{in} and e''_{out} in counterclockwise order is denoted by $A^r_{NS}(v)$. The set of the remaining angles at v is denoted by $A^m_{NS}(v)$.

Variables. We associate a variable ℓ_a with each angle $a = (e_1, v, e_2)$. Variable ℓ_a takes values 0, 1, or 2, which correspond to the labels S , F , or L for a , respectively.

For each edge $e = (u, v)$ we define two variables, o_{uv} and o_{vu} , that take values in the set $\{0, 1\}$ and that define the orientation of e in G' ; if $o_{uv} = 1$ edge e is oriented from u to v , otherwise it is oriented from v to u .

Finally, for each angle $a = (e_1, v, e_2)$, we define a variable c_a that takes values in the set $\{-1, 0, 1\}$. This variable is used to guarantee consistency between the orientations of e_1, e_2 and the value of ℓ_a , as explained later.

Constraints. We must guarantee the properties of Theorem 1. For an internal face (resp. the external face) f of G , denote by $cap(f)$ the number of angles in f minus 2 (resp. plus 2). Properties (a) and (b) are guaranteed by the following constraints:

$$\sum_{a \in A(f)} \ell_a = cap(f), \quad \forall f \in F. \tag{1}$$

Consistency about the orientations of the edges is ensured by Constraints 2: The first constraint forces the directed edges of G to keep their orientation in G' , and the second avoids that an edge can receive two distinct orientations at the same time.

$$o_{uv} = 1, \quad \forall (u, v) \in E_D, \quad o_{uv} + o_{vu} = 1, \quad \forall (u, v) \in E. \tag{2}$$

Properties (c) – (e) are guaranteed by Constraints 3 and 4.

$$\sum_{a \in A_{PS}(v)} \ell_a = 2, \quad \forall v \in V_{PS} \tag{3}$$

$$\sum_{a \in A^l_{NS}(v)} \ell_a = 1, \quad \sum_{a \in A^r_{NS}(v)} \ell_a = 1, \quad \sum_{a \in A^m_{NS}(v)} \ell_a = 0, \quad \forall v \in V_{NS} \tag{4}$$

Finally, for each angle $a = (e_1, v, e_2)$ we have to guarantee consistency between its label and the orientation of the edges e_1 and e_2 . Namely, denote by v_1 the vertex of e_1 other than v , and denote by v_2 the vertex of e_2 other than v . If o_{vv_1} and o_{vv_2} have the same value (which means that e_1 and e_2 are both incoming or both outgoing v) then ℓ_a

must take a value in $\{0, 2\}$. Otherwise, ℓ_a must take value 1. This property is forced by the following constraint:

$$o_{vv_1} + o_{vv_2} = \ell_a + 2c_a, \quad \forall a \in A \quad (5)$$

We observe that Constraint 5 and the integrality constraints on variables o_{uv} and c_a , imply that variables ℓ_a always assume integer values. Hence, we can relax the integrality constraints on ℓ_a , by simply requiring that $0 \leq \ell_a \leq 2$. Also, note that the total number of variables and constraints of our model is linear in the number of angles and edges of G ; therefore, since G is planar, it is linear in the number of vertices of G . The next theorem summarizes the main contribution of this section.

Theorem 2. *There exists an ILP model to decide if an embedded mixed planar graph admits an upward planar embedding, and to find one in the positive case. The number of variables and constraints of the model is linear in the number of vertices of the graph.*

4 Experimental Study

We implemented our ILP model using CPLEX and we experimented it on a large set of mixed graphs, in order to understand if it is computationally feasible in practice. We focused on two major issues: (i) What is the time required to find an upward planar embedding of an embedded mixed graph, if there exists one; (ii) what is the time required to decide whether a mixed graph admits or not an upward planar embedding. To this aim, we run the experiments on two different test suites of mixed graphs, which we refer to as MIXEDPOSITIVE and MIXEDGENERAL. MIXEDPOSITIVE contains mixed embedded planar graphs that always admit an embedding-preserving upward planar drawing. Hence, for these graphs the computation will never reject the instance, and we can measure the time required to find an upward planar embedding. MIXEDGENERAL contains mixed embedded planar graphs for which an upward planar drawing may or may not exist. From the experiments we expect that the computation is faster on those instances that do not admit a solution and that on the positive instances the time required to find an embedding increases when the number of undirected edges increases.

Each graph G in MIXEDPOSITIVE was generated by first generating an upward planar embedded digraph G' with the algorithm described in [4], and then removing the orientation on a certain percentage of edges of G' . The edges that are made undirected were selected randomly with a uniform probability distribution. Each graph G in MIXEDGENERAL was generated with the following procedure: Again, we first generated an upward planar embedded digraph G' with the algorithm in [4]. Then a planar embedded mixed graph was computed from G' by repeating the following steps until the desired percentage of undirected edges was reached: randomly choose a face f of G' and add an edge in f randomly selecting its end-vertices (multiple edges were avoided); then randomly remove from G' a directed edge, while maintaining the connectivity. Every random choice followed a uniform probability distribution. Set MIXEDPOSITIVE contains 3 graphs for each distinct triple $\langle n, d, p \rangle$, where $n \in \{100, 200, \dots, 800\}$ is the number of vertices, $d \in \{1.4, 1.6, 1.8, 2.0\}$ is the density, and $p \in \{20, 50, 80\}$ is the percentage of undirected edges of the graph. Hence, MIXEDPOSITIVE contains

288 graphs in total. Set MIXEDGENERAL contains 10 graphs for each distinct triple $\langle n, d, p \rangle$, where n , d , and p take the same values as before. Hence, it contains 960 graphs in total.

The experiments were performed under the Windows Vista OS, on an Intel Core-Duo with 2.2 GHz and 2 GB of RAM; the computations were rather fast and confirmed our hypothesis. As expected, the CPU time for the graphs in MIXEDPOSITIVE increases when the percentage of undirected edges increases. Almost all computations required less than 4 seconds, and the maximum time of a computation was 12 seconds, for an instance with 600 vertices, 80% of undirected edges, and density 2.0.

The percentage of negative instances in MIXEDGENERAL (i.e., the percentage of graphs for which an upward planar embedding does not exist) is close to 100% for most graphs with no more than 50% of directed edges, while about half of the graphs with 80% of undirected edges admit a solution. As expected, the computation is very fast on the negative instances, while the behavior on the positive instances reflects the one for the graphs in MIXEDPOSITIVE.

5 Conclusions and Open Problems

We introduced a new upward planarity testing problem for embedded mixed graphs and we experimentally showed that this problem can be efficiently solved using Integer Linear Programming. The main open problem is to study what is the computational complexity of our testing problem in theory. Is it NP-hard? It is worth recalling that the upward planarity testing problem for embedded digraphs is polynomially solvable [2] and that polynomial-time algorithms exist for finding upward embeddings of embedded undirected graphs [6]. Another interesting problem is to design algorithmic solutions for computing the maximum upward planar subgraph for a given mixed embedded graph. We recall that the problem of computing a maximum upward planar subgraph of a planar embedded digraph is NP-hard [3].

References

1. Bang-Jensen, J., Gutin, G.: *Digraphs: Theory, Algorithms and Applications*, 2nd edn. Springer, Heidelberg (2009)
2. Bertolazzi, P., Di Battista, G., Liotta, G., Mannino, C.: Upward drawings of triconnected digraphs. *Algorithmica* 6(12), 476–497 (1994)
3. Binucci, C., Didimo, W., Giordano, F.: Maximum upward planar subgraphs of embedded planar digraphs. *Comput. Geom.* 41(3), 230–246 (2008)
4. Didimo, W.: Upward planar drawings and switch-regularity heuristics. *Journal of Graph Algorithms and Applications* 10(2), 259–285 (2006)
5. Didimo, W., Giordano, F., Liotta, G.: Upward spirality and upward planarity testing. *SIAM J. Discrete Math.* 23(4), 1842–1899 (2009)
6. Didimo, W., Pizzonia, M.: Upward embeddings and orientations of undirected planar graphs. *Journal of Graph Algorithms and Applications* 7(2), 221–241 (2003)
7. Eiglsperger, M., Eppinger, F., Kaufmann, M.: An approach for mixed upward planarization. *Journal of Graph Algorithms and Applications* 7(2), 203–220 (2003)
8. Garg, A., Tamassia, R.: Upward planarity testing. *Order* 12, 109–133 (1995)
9. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing* 31(2), 601–625 (2001)