Università degli Studi di Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

Upward Embeddings and Orientations of Undirected Planar Graphs

Walter Didimo[1], Maurizio Pizzonia[1]

RT-DIA-65-2001 **April 2001**

ABSTRACT

An *upward embedding* of an embedded planar graph states, for each vertex v, which edges are incident to v "above" or "below" and, in turn, induces an *upward orientation* of the edges. In this paper we characterize the set of all upward embeddings and orientations of a plane graph by using a simple flow model. We take advantage of such a flow model to compute upward orientations with the minimum number of sources and sinks of 1-connected graphs. Our theoretical results allow us to easily compute visibility representations of 1-connected graphs while having a certain control over the width and the height of the computed drawings, and to deal with partial assignments of the upward embeddings "underlying" the visibility representations.

1 Introduction

Let G be an undirected planar graph with a given planar embedding. Loosely speaking, an *upward embedding* (also called an *upward representation*) of G is given by splitting, for each vertex v of G, the ordered circular list of the edges that are incident to v into two linear lists $L_{above}(v)$ and $L_{below}(v)$, in such a way that there exists a planar drawing Γ of G with the following properties: (i) all the edges are monotonically increasing in the vertical direction; (ii) for each vertex v the edges in $L_{above}(v)$ ($L_{below}(v)$) are incident to v above (below) the horizontal line through v. Drawing Γ is said to be an *upward drawing* of G. An orientation of all edges of Γ from bottom to top defines an orientation of all edges of G, which we call an *upward orientation* of G. Hence, any upward embedding of G induces an upward orientation of G. Figure 1 shows an upward embedding of a plane graph and the upward orientation induced by it.

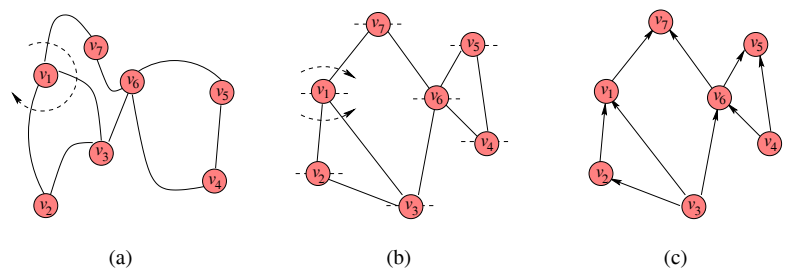


Figure 1: (a) An embedded planar graph. (b) An upward embedding of the embedded planar graph. For each vertex v_i of the graph the edges in $L_{below}(v_i)$ and $L_{above}(v_i)$ are drawn incident below and above the horizontal line through v_i, respectively. (c) The upward orientation induced by the upward embedding.

Upward embeddings and orientations of undirected graphs have been widely studied within specific theoretical and application domains. For example, a deep investigation of the properties of upward embeddings and drawings for ordered sets and planar lattices can be found in [14, 13, 6, 17]. Relations between the problem of finding bar layout drawings of weighted undirected graphs and the problem of computing upward orientations with specific properties are provided in [15, 19]. An important class of upward orientations is represented by the so called *bipolar orientations* (or *st-orientations*). A bipolar orientation of an undirected planar graph G is an upward orientation of G with exactly one source s (vertex without in-edges) and one sink t (vertex without out-edges). A bipolar orientation of G with source s and sink t exists if and only if $G \cup \{(s,t)\}$ is biconnected. Finding a bipolar orientation of a planar graph is often the first step of many algorithms in graph theory and graph drawing. The properties of bipolar orientations have been extensively studied in [7], and a characterization of bipolar orientations in terms of a network flow model is described in [5].

Many results on upward embeddings of digraphs have been also provided in the literature. In this case, the orientation of the edges of the graph is given, and a classical problem consists in finding a planar upward embedding within such an orientation. Clearly, a planar upward embedding of a digraph might not exist. In [3] a polynomial time algorithm for testing the existence of planar upward embeddings of a digraph within a given embedding is described. The algorithm is also able to construct an upward embedding if there exists

one. In the variable embedding the upward planarity testing problem is NP-complete [10], but it can be solved in polynomial time for digraphs with a single source [4].

In this paper we focus on upward embeddings and orientations of undirected planar graphs. The main contributions of our work are the following:

- Starting from the properties on upward planarity given in [3], we provide a full characterization of the set of all upward embeddings and orientations of any embedded planar graph (Section 3.1). It is based on a network flow model, which is related to the one used in [5] for characterizing bipolar orientations. In particular, if the graph is biconnected, our flow model also captures all bipolar orientations of the graph.

- We describe flow based polynomial time algorithms for computing upward embeddings of the input graph. Such algorithms allow us to deal with partial assignments of the upward embedding (Section 3.1). Further, we provide a polynomial time algorithm to compute upward orientations with the minimum number of sources and sinks (Section 3.2). An upward orientation with the minimum number of sources and sinks can be viewed as a natural extension of the concept of bipolar orientation to 1-connected graphs.

- We describe a simple technique to compute visibility representations of 1-connected planar graphs (Section 4), which can be of practical interest for graph drawing applications. It is based on the computation of an upward embedding of the graph, and does not require running any augmentation algorithm to initially make the graph biconnected. Compared to a standard technique that uses the approximation algorithm in [9] to make the graph biconnected, the algorithm we propose is faster and achieves similar results in terms of area of the visibility representation. We also present (Section 4.1) a preliminary experimental study that shows how our technique can be used to have a certain control over the width and the height of the visibility representations.

In Section 2 we give some basic definitions and results on upward embeddings and orientations of undirected planar graphs.

2 Basic Definitions and Results on Upward Embeddings

Let G be a graph. A *drawing* Γ of G maps each vertex u of G into a point p_u of the plane and each edge (u, v) of G into a Jordan curve between p_u and p_v. Γ is *planar* if two distinct edges never intersect except at common end-points. G is *planar* if it admits a planar drawing. A planar drawing Γ of G divides the plane into topologically connected regions called *faces*. Exactly one of these faces is unbounded, and it is said to be *external*; the others are called *internal* faces. Also, for each vertex v of G, Γ induces a circular clockwise ordering of the edges incident on v. The choice ϕ of such an ordering for each vertex of G and of an external face is called a *planar embedding* of G. A planar graph G with a given planar embedding ϕ is called an *embedded planar* graph and denoted by G_ϕ. A *drawing* of G_ϕ is a planar drawing of G that induces ϕ as the planar embedding.

Let G_ϕ be an (undirected) embedded planar graph. An *upward embedding* $\mathcal{E}_\phi$ of G_ϕ is a splitting of the adjacency lists of all vertices of G_ϕ such that:

`Pr.a` For each vertex v of G_ϕ the circular clockwise list $L(v)$ of the edges incident on v is split into two linear lists, $L_{below}(v)$ and $L_{above}(v)$, so that the circular list obtained by concatenating $L_{above}(v)$ and the reverse of $L_{below}(v)$ is equal to $L(v)$.

`Pr.b` There exists a planar drawing $\Gamma(\mathcal{E}_\phi)$ of G_ϕ such that all the edges are monotonically increasing in the vertical direction and for each vertex v of G_ϕ the edges of $L_{below}(v)$ and $L_{above}(v)$ are incident to v below and above the horizontal line through v, respectively. We say that $\Gamma(\mathcal{E}_\phi)$ is a *drawing* of $\mathcal{E}_\phi$ and an *upward drawing* of G_ϕ.

From `Pr.b` the following is immediate.

Property 1 *Given an upward embedding of G_ϕ, for each edge $e = (u, v)$ of G_ϕ either $e \in L_{above}(u) \cap L_{below}(v)$ or $e \in L_{below}(u) \cap L_{above}(v)$.*

An upward embedding $\mathcal{E}_\phi$ of G_ϕ uniquely induces an *upward orientation* $\mathcal{O}_\phi$ of G_ϕ. Namely, for each edge $e = (u, v)$ such that $e \in L_{above}(u)$ and $e \in L_{below}(v)$, we orient e from u to v (see Figure 1). Conversely, an upward orientation defines in general a class of possible upward embeddings inducing that orientation (see Figure 2). A *source* of $\mathcal{E}_\phi$ is a vertex v of G_ϕ such that $L_{below}(v)$ is empty. A source has only out-edges with respect to orientation $\mathcal{O}_\phi$. A *sink* of $\mathcal{E}_\phi$ is a vertex v of G_ϕ such that $L_{above}(v)$ is empty. A sink has only in-edges with respect to $\mathcal{O}_\phi$.

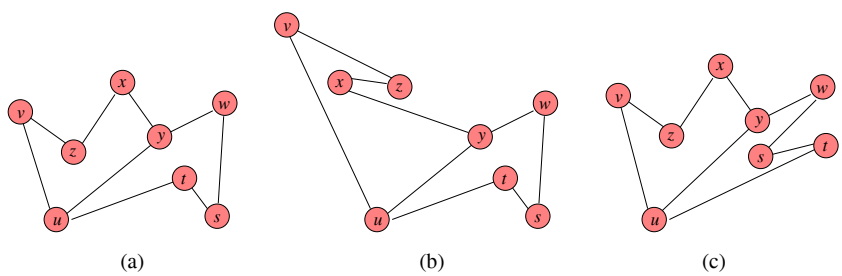


Figure 2: Three different upward embeddings that induce the same upward orientation.

Given a vertex v of G_ϕ, we denote by $deg(v)$ the number of edges incident on v. An *angle* of G_ϕ at vertex v is a pair of clockwise consecutive edges incident on v. In particular, if $deg(v) = 1$, and if we denote by e the edge incident on v, $\{e, e\}$ is an angle. Given a splitting of the adjacency lists of G_ϕ that verifies `Pr.a`, an angle $\{e_1, e_2\}$ at vertex v of G_ϕ can be of three types:

- *large*: (i) both e_1 and e_2 belong to $L_{below}(v)$ ($L_{above}(v)$), and (ii) e_1 and e_2 are the first (last) edge and the last (first) edge of $L_{below}(v)$ ($L_{above}(v)$), respectively. We associate a label L with a large angle.
- *flat*: if: (i) $e_1 \in L_{below}(v)$ and $e_2 \in L_{above}(v)$ or, (ii) $e_1 \in L_{above}(v)$ and $e_2 \in L_{below}(v)$. We associate a label F with a flat angle.
- *small*: in all the other cases. We associate a label S with a small angle.

Figure 3 shows the labeling of the angles of a graph G_ϕ determined by an upward embedding $\mathcal{E}_\phi$. Each drawing of $\mathcal{E}_\phi$ maps the angles of G_ϕ to geometric angles such that large and small angles always correspond to geometric angles larger and smaller than 180 degrees, respectively. Both the two edges that form a large or a small angle at vertex v are incident to v either above or below the horizontal line through v. Instead, a flat angle at vertex v corresponds to a geometric angle that can be either larger or smaller than 180 degrees; in any case, an edge of the angle is incident to v above the horizontal line through v while the other edge is incident to v below the same line.

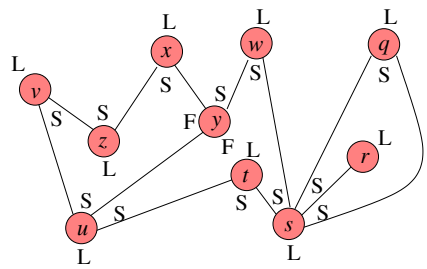


Figure 3: Labeling of the angles of an embedded planar graph determined by an upward embedding of the graph.

Let f be a face of G_ϕ. We call *border* of f the alternating circular list of the vertices and edges that form the boundary of f. Note that, if the graph is not biconnected an edge or a vertex may appear more than once in the border of f. We say that an angle $\{e_1, e_2\}$ at vertex v *belongs* to face f if e_1, e_2, and v belong to the border of f. The *degree* of f, denoted by $deg(f)$, is the number of edges in the border of f. Observe that, $deg(f)$ is equal to the number of angles of f .

Consider now any labeling of the angles of G_ϕ with labels L, S, and F. For each face f of G_ϕ denote by $L(f)$, $S(f)$, and $F(f)$ the number of angles that belong to f with label L, S, and F, respectively. Also, for each vertex v of G_ϕ denote by $L(v)$, $S(v)$, and $F(v)$ the number of angles at vertex v with label L, S, and F, respectively. The following lemma is a restatement of a known result on upward planarity [3].

Lemma 1 *Let $\mathcal{E}_\phi$ be a splitting of the adjacency lists of G_ϕ that verifies* `Pr.a`*, and consider the labeling of the angles of G_ϕ determined by it. $\mathcal{E}_\phi$ is an upward embedding of G_ϕ if and only if the following properties hold:*

(a) $S(f) = L(f) + 2$*, for each internal face f of G_ϕ.*

(b) $S(f) = L(f) - 2$*, for the external face f of G_ϕ.*

(c) $F(v) = 2$*, $S(v) = deg(v) - 2$, and $L(v) = 0$, for each vertex v of G_ϕ such that both $L_{above}(v)$ and $L_{below}(v)$ are not empty.*

(d) $F(v) = 0$*, $S(v) = deg(v) - 1$, and $L(v) = 1$, for each vertex v of G_ϕ such that either $L_{above}(v)$ or $L_{below}(v)$ is empty.*

Properties (c) and (d) of Lemma 1 state that if $\mathcal{E}_\phi$ is an upward embedding of G_ϕ, each source or sink of $\mathcal{E}_\phi$ has exactly one large angle and no flat angles, while each vertex that is neither a source nor a sink has exactly two flat angles and no large angles. The next lemma provides a different formulation for properties (a) and (b).

Lemma 2 *Properties* (a) *and* (b) *of* Lemma 1 *are equivalent to the following properties:*

(a') $deg(f) - 2 = 2L(f) + F(f)$, *for each internal face* f *of* G_ϕ.

(b') $deg(f) + 2 = 2L(f) + F(f)$, *for the external face* f *of* G_ϕ.

Proof: Since $deg(f)$ is equal to the number of angles of f, from property (a) we have that: $deg(f) = L(f) + S(f) + F(f) = L(f) + L(f) + 2 + F(f) = 2L(f) + 2 + F(f)$, and then $deg(f) - 2 = 2L(f) + F(f)$. Hence, (a) implies (a'). Conversely, substituting $deg(f)$ with $L(f) + S(f) + F(f)$ in the equality $deg(f) - 2 = 2L(f) + F(f)$, we get property (a). Hence, (a') implies (a). Therefore (a') and (a) are equivalent. Analogously, we can prove that property (b') is equivalent to property (b). *q.e.d.*

3 Characterizing Upward Embeddings

In this section we give a full characterization of the set of all upward embeddings of a general embedded planar graph (Section 3.1). This also implies a characterization of all upward orientations of the given graph. Our characterization uses a flow model that is related to the one described in [5] for bipolar orientations. Also, we show how it is possible to add costs to our flow model in order to compute in polynomial time an upward orientation with the minimum number of sources and sinks (Section 3.2).

3.1 A Flow Model for Characterizing Upward Embeddings

The following theorem characterizes the class of labelings that are determined by all upward embeddings of an embedded planar graph. It is important to observe that the characterization of such a class of labelings does not depend either on the choice of a splitting of the adjacency lists of the graph, in contrast to the result given in Lemma 1, or on the choice of an orientation of the graph.

Theorem 1 *Let* $\mathcal{L}$ *be any labeling of the angles of an embedded graph* G_ϕ *with labels* L, S, *and* F. $\mathcal{L}$ *is the labeling determined by an upward embedding of* G_ϕ *if and only if the following properties hold:*

(a') $deg(f) - 2 = 2L(f) + F(f)$, *for each internal face* f *of* G_ϕ.

(b') $deg(f) + 2 = 2L(f) + F(f)$, *for the external face* f *of* G_ϕ.

(c') *For each vertex* v *either* $F(v) = 2$ *and* $L(v) = 0$ *or* $F(v) = 0$ *and* $L(v) = 1$.

Proof:

The necessary condition is an immediate consequence of Lemma 1 and Lemma 2. In fact, if $\mathcal{L}$ is determined by an upward embedding, then properties (a), (b), (c), and (d) of Lemma 1 are verified. From Lemma 2 properties (a) and (b) are equivalent to properties

(a') and (b'); further, properties (c) and (d) imply that one of the two cases of property (c') holds, for each vertex of G_ϕ.

To prove the sufficiency of the condition, we consider a labeling $\mathcal{L}$ that verifies properties (a'), (b'), and (c'), and construct an upward embedding of G_ϕ that determines $\mathcal{L}$. From $\mathcal{L}$, we construct a splitting $\mathcal{E}_\phi$ of the adjacency lists of G_ϕ as follows:

- We observe that there exists at least two distinct vertices s and t on the external face h having an angle labeled with L. In fact, from property (b') (that is equivalent to property (b) of Lemma 1) we must have that $L(h) = S(h) + 2$. We assign all the edges incident on s to the list $L_{above}(s)$ (we set $L_{below}(s)$ empty). Namely, denoted by $\{e_1, e_2\}$ the angle at vertex s with label L, e_2 and e_1 will be the first edge and the last edge of $L_{above}(s)$, respectively.

- We execute a breadth first search starting from s. At each step we visit a different vertex v and split the list of the edges that are incident on v. Namely, suppose that v is visited by moving from vertex u through edge $e_0 = (u, v)$ (e_0 is the father edge of v in the breadth first search). If e_0 is in $L_{above}(u)$ we put e_0 in $L_{below}(v)$, while if e_0 is in $L_{below}(u)$ we put e_0 in $L_{above}(v)$. Suppose that $e_0, e_1, \ldots, e_k$ are the edges incident on v in this clockwise ordering. For each e_i ($i = 0, \ldots, k-1$) we consider the label l of angle $\{e_i, e_{i+1}\}$, and we decide if e_{i+1} has to be assigned to $L_{above}(v)$ or to $L_{below}(v)$. Note that, at this point, e_i has been already assigned to one of the two lists. The following cases are possible: (1) If $l = L$ and $e_i \in L_{below}(v)$ then e_{i+1} is put at the end of $L_{below}(v)$. (2) If $l = L$ and $e_i \in L_{above}(v)$ then e_{i+1} is put at the start of $L_{above}(v)$. (3) If $l = S$ and $e_i \in L_{below}(v)$ then e_{i+1} is put immediately before e_i in $L_{below}(v)$. (4) If $l = S$ and $e_i \in L_{above}(v)$ then e_{i+1} is put immediately after e_i in $L_{above}(v)$. (5) If $l = F$ and $e_i \in L_{below}(v)$ then e_{i+1} is put at the start of $L_{above}(v)$. (6) If $l = F$ and $e_i \in L_{above}(v)$ then e_{i+1} is put at the end of $L_{below}(v)$.

It is easy to see that $\mathcal{E}_\phi$ verifies `Pr.a`. To prove that $\mathcal{E}_\phi$ is an upward embedding of G_ϕ we can just prove that properties (c) and (d) of Lemma 1 are verified (in fact properties (a) and (b) are equivalent to properties (a') and (b')). From property (c') we only have two possible cases for the labels of the angles at each vertex v of G_ϕ.

- $F(v) = 2$ and $L(v) = 0$. This implies that, for splitting the edges incident on v cases (1) and (2) are never applied, cases (5) and (6) are applied twice in the total, and cases (3) and (4) are applied $deg(v) - 2$ times in the total. Also, cases (5) and (6) imply that neither $L_{above}(v)$ nor $L_{below}(v)$ will be empty. This matches property (c).

- $F(v) = 0$ and $L(v) = 1$. This implies that, for splitting the edges incident on v, either case (1) or case (2) is applied once, cases (5) and (6) are never applied, and either case (3) or case (4) is applied $deg(v) - 1$ times. Also, observe that both cases (1), (2), (3), and (4) always put e_{i+1} in the same list as e_i, and that either (1) and (3) or (2) and (4) are applied. This guarantees that exactly one of the two lists $L_{above}(v)$ and $L_{below}(v)$ will be empty. This matches property (d).

Finally, since no other cases are possible, properties (c) and (d) of Lemma 2 are verified.

q.e.d.

We call *upward labeling* of G_ϕ a labeling of the angles of G_ϕ that verifies properties (a'), (b'), and (c') of Theorem 1. The result of Theorem 1 allows us to describe all upward embeddings of G_ϕ by considering all upward labelings of G_ϕ. Note that, the proof of the theorem gives a method to construct the upward embedding associated with an upward labeling. Actually, for each upward labeling, there are exactly two "symmetric" upward embeddings that determine it; they are obtained one from the other by simply exchanging list $L_{above}(v)$ with list $L_{below}(v)$ for each vertex v and then reversing such lists (see Figure 5 (b)).

We now provide a network flow model that characterizes all the upward labelings of G_ϕ. Because of the above considerations, this flow model provides a characterization of all upward embeddings of G_ϕ. We associate with G_ϕ a flow network $\mathcal{N}_\phi$, such that the integer feasible flows on $\mathcal{N}_\phi$ are in one-to-one correspondence with the upward labelings of G_ϕ. Flow network $\mathcal{N}_\phi$ is a directed graph defined as follows (see Figure 4):

- The nodes of $\mathcal{N}_\phi$ are the vertices (*vertex-nodes*) and the faces (*face-nodes*) of G_ϕ. Each vertex-node supplies flow 2 and each face-node associated with face f of G_ϕ demands a flow equal to $deg(f) - 2$ if f is internal and $deg(f) + 2$ if f is external.

- For each angle of G_ϕ at vertex v in face f there is an associated arc (v, f) of $\mathcal{N}_\phi$ with lower capacity 0 and upper capacity 2.

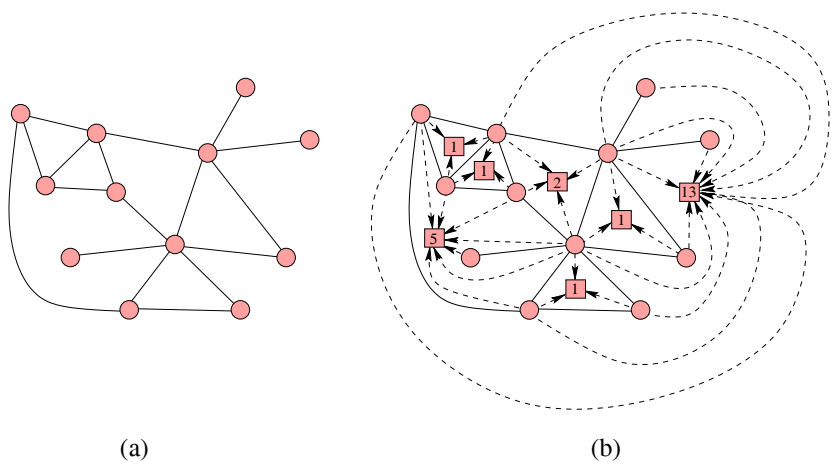


Figure 4: (a) An embedded planar graph G_ϕ. (b) Flow network $\mathcal{N}_\phi$ associated with G_ϕ. The vertex-nodes are circles and the face-nodes are squares. Each face-node is marked with its demand. The arcs of the networks are dashed.

Observe that in $\mathcal{N}_\phi$ the total demand is equal to the total supply. In fact: $\sum_{f \in F} (deg(f) - 2) + 4 = \sum_{f \in F} deg(f) - 2|F| + 4 = 2|E| - 2|F| + 4 = 2|V|$. The intuitive interpretation of the flow model in terms of upward embedding is as follows: (i) Each unit of flow represents a flat angle, with the convention that a large angle counts as two flat angles; an arc a of $\mathcal{N}_\phi$ has flow $0, 1$, or 2, depending on the fact that its associated angle is small, flat, or large, respectively. (ii) The demand of each face-node and the supply of each vertex-node reflect the balancing properties (a'), (b') and (c'). Figure 5 shows a feasible flow on the network associated with an embedded planar graph, the corresponding upward labeling,

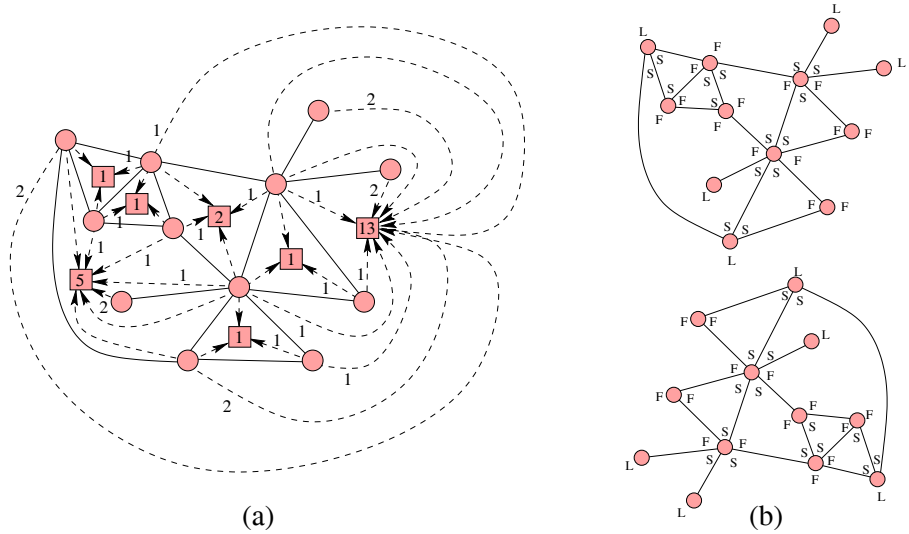


Figure 5: (a) A feasible flow on the network associated with an embedded planar graph. Only the flow values different from zero are shown. (b) The upward labeling $\mathcal{L}$ corresponding to the flow and the two "symmetric" upward embeddings associated with $\mathcal{L}$.

and the two "symmetric" upward embeddings associated with the labeling. Theorem 2 formally proves the correctness of the intuitive interpretation above described.

The flow network used in [5] to characterize the bipolar orientations of an embedded planar graph is tailored for biconnected planar graphs and captures only bipolar orientations. The values of the flow are not able to represent large angles (the flow values are only 0 or 1), except for the source and the sink of the orientation. Our flow network generalizes this network to represent any kind of upward orientations and embeddings, including the bipolar orientations.

Theorem 2 *Let G_ϕ be an embedded planar graph and let $\mathcal{N}_\phi$ be the flow network associated with G_ϕ. There is a one-to-one correspondence between the set of the upward labelings of G_ϕ and the set of the integer feasible flows on $\mathcal{N}_\phi$.*

Proof:

Consider an upward labeling $\mathcal{L}$ of G_ϕ. From it we construct an integer feasible flow x of $\mathcal{N}_\phi$ as follows. For each angle α of G_ϕ let a be the arc of $\mathcal{N}_\phi$ associated with α. We set $x(a) = 2$ if α is labeled L, $x(a) = 1$ if α is labeled F, and $x(a) = 0$ if α is labeled S. The above construction is clearly an injective transformation. In fact, there is a one-to-one correspondence between angles of G_ϕ and arcs of $\mathcal{N}_\phi$ and hence, different labelings of the same angle of G_ϕ produces different values of flow on the corresponding arc of $\mathcal{N}_\phi$. We now prove that flow x is feasible. From the construction of x and from property (c') of $\mathcal{L}$, it follows that every vertex-node of $\mathcal{N}_\phi$ supplies flow 2 (and demands flow 0). Hence, the balance property of x on every vertex-node of $\mathcal{N}_\phi$ is verified. Let f be an internal (the external) face of G_ϕ, and consider the face-node of $\mathcal{N}_\phi$ associated with f. From the construction of x, such a face-node receives a flow equal to $2L(f) + F(f)$ and supplies flow 0 ; hence, from property (a') ((b')) of $\mathcal{L}$, it demands a flow equal to $deg(f) - 2$ ($deg(f) + 2$). Hence, also the balance property of x on every face-node is verified. Finally,

since on each arc of $\mathcal{N}_\phi$ we assign an integer amount of flow in the range $[0, 2]$, the lower and upper capacities on the arcs of $\mathcal{N}_\phi$ are respected by x.

Conversely, consider an integer feasible flow x of $\mathcal{N}_\phi$, and construct from x a labeling $\mathcal{L}$ of G_ϕ, by applying a transformation that is the reverse of that above described. Namely, for each arc a of $\mathcal{N}_\phi$ denote by α the corresponding angle of G_ϕ. Labeling $\mathcal{L}$ is constructed by assigning label L, F, and S to α, depending on the case that $x(a) = 2$, $x(a) = 1$, and $x(a) = 0$, respectively. By using the properties of x and the same reasoning applied above, it is easy to prove that $\mathcal{L}$ is an upward labeling of G_ϕ. *q.e.d.*

Theorem 1 and Theorem 2 allow us to compute an upward embedding of an embedded planar graph G_ϕ, by computing an integer feasible flow on network $\mathcal{N}_\phi$. Denote by n the number of vertices of G_ϕ. Since network $\mathcal{N}_\phi$ is planar and has $O(n)$ vertices, a feasible flow on $\mathcal{N}_\phi$ can be computed in $O(n \log n)$ time by applying a known maximum flow algorithm for planar networks [1]. Also, both $\mathcal{N}_\phi$ and an upward embedding associated with a feasible flow on $\mathcal{N}_\phi$ can be constructed in linear time. Therefore, an upward embedding of any embedded planar graph can be constructed in $O(n \log n)$ with the above technique.

We remark that there are two main advantages of computing upward embeddings of a general plane graph G_ϕ by using the flow model described so far: No augmentation algorithms have to be used to initially biconnected the graph (we just apply a standard flow algorithm); it is possible to fix the flow on some arcs of the network to constrain the upward embedding to have a partially specified "shape". For example, we can specify that an angle must be large in a certain face, or that some vertices must be neither sources nor sinks. In the next section we describe how to compute upward embeddings with the minimum number of sources and sinks, by adding costs to our network.

3.2 Minimizing Sources and Sinks

Computing an upward embedding of G_ϕ with the minimum number of sources and sinks (which we call *optimal upward embedding* for simplicity) is equivalent to computing an upward embedding with the minimum number of large angles. Clearly, if the graph is biconnected, the problem is reduced to the computation of a bipolar orientation. For this reason, we regard the concept of optimal upward orientation as the natural extension of the definition of bipolar orientation to the case of general connected graphs.

The flow model we use to compute an optimal upward orientation of G_ϕ is a variation of the one described for characterizing upward embeddings (see Section 3.1). We add a linear number of arcs to network $\mathcal{N}_\phi$ and we equip the arcs of the new network with costs. Each unit of cost represents a large angle. We also reduce the upper capacity of all the arcs of the network. More in detail, we define a network $\mathcal{N}'_\phi$ as follows:

- The nodes of $\mathcal{N}'_\phi$ are again the vertices (vertex-node) and the faces (face-nodes) of G_ϕ. Each vertex-node again supplies flow 2 and each face-node associated with face f of G_ϕ again demands flow $deg(f) - 2$ if f is internal and $deg(f) + 2$ if f is external.

- For each angle of G_ϕ at vertex v in face f there is an associated pair of directed arcs $a_v = (v, f), a'_v = (v, f)$ in $\mathcal{N}'_\phi$. Both the arcs have lower capacity 0 and upper capacity 1. Also, arc a_v has cost 0 while arc a'_v has cost 1.

In $\mathcal{N}'_\phi$ we compute a minimum cost flow x. The interpretation of the flow in terms of upward labeling is similar to the one given for $\mathcal{N}_\phi$, with a slightly variation due to

the additional arcs and costs. We first observe that for each pair of arcs a_v, a'_v it never happens $x(a_v) = 0$ and $x(a'_v) = 1$, due to the fact that the cost of a_v is 0 and the cost of a'_v is 1. In fact, if $x(a_v) = 0$ and $x(a'_v) = 1$, then there would exist a negative cost cycle represented by the two arcs a'_v, a_v, and it would be possible to derive a new flow x' from x by simply exchanging one unit of flow between a'_v and a_v (i.e., $x'(a_v) = 1$ and $x'(a'_v) = 0$). This would imply that x' has a cost smaller than the cost of x, in contrast to the assumption that x has the minimum cost. Hence, the only possibilities for the flow on arcs a_v, a'_v are: (i) $x(a_v) = x(a'_v) = 0$, the angle associated with arcs a_v, a'_v is small. (ii) $x(a_v) = 1$ and $x(a'_v) = 0$, the angle associated with arcs a_v, a'_v is flat. (iii) $x(a_v) = x(a'_v) = 1$, the angle associated with arcs a_v, a'_v is large.

Note that, only in the third case we have cost 1 on arcs a_v, a'_v, while in the other two cases we have cost 0. This implies that the total cost of flow x on $\mathcal{N}'_\phi$ represents the total number of large angles of the corresponding upward embedding of G_ϕ. Hence, since x has the minimum cost, the corresponding upward embedding has the minimum number of large angles.

Let n be the number of vertices of G_ϕ. Since network $\mathcal{N}'_\phi$ is planar and has $O(n)$ vertices, and since its total demand (supply) is $O(n)$, a minimum cost flow on $\mathcal{N}'_\phi$ can be computed in $O(n^{\frac{7}{4}} \log n)$ time by the algorithm described in [11].

We conclude this section by giving an upper bound on the number of sources and sinks of an optimal upward embedding.

Lemma 3 *An optimal upward embedding of an embedded planar graph G_ϕ has at most $B + 1$ sources and sinks, where B is the number of blocks of G_ϕ. Also, in the optimal upward embedding each block contains at most one source and one sink.*

Proof: We prove the lemma by induction on B. If $B = 1$, the graph is biconnected and an optimal upward embedding of it has exactly one source and one sink. Suppose that the lemma is true for each graph with $B \geq 2$ blocks, and consider a graph G_ϕ with $B + 1$ blocks. Let $G'_{\phi'}$ be the graph obtained from G_ϕ by removing any block C with exactly one cutvertex v, and let $\mathcal{E}'_{\phi'}$ be an optimal upward embedding of $G'_{\phi'}$. From the inductive hypothesis, $\mathcal{E}'_{\phi'}$ has at most $B + 1$ sources and sinks. From $\mathcal{E}'_{\phi'}$ we construct an upward embedding of G_ϕ. Such an upward embedding coincides with $\mathcal{E}'_{\phi'}$ for the subgraph $G'_{\phi'}$ and it is determined on C as follows: If C has to be attached above v in $\mathcal{E}'_{\phi'}$, we compute an upward embedding of C with exactly one source and one sink, where the source is vertex v. If C has to be attached below v in $\mathcal{E}'_{\phi'}$, we compute an upward embedding of C with exactly one source and one sink, where the sink is v. The obtained upward embedding has either one source or one sink more than $\mathcal{E}'_{\phi'}$, since vertex v is in common between C and $G'_{\phi'}$. Also, the iterative construction used to prove the induction leads to upward embeddings in which each block contains at most one source and one sink. *q.e.d.*

The bound of Lemma 3 is strict and a class of plane graphs whose upward embeddings have $B+1$ sources and sinks can be obtained by nesting each block into another, as shown by the example of Figure 6.

4 Algorithms for Visibility Representations

We use the above results on upward embeddings to compute drawings of general connected planar graphs. Namely, we focus on graph drawing algorithms which require the compu-

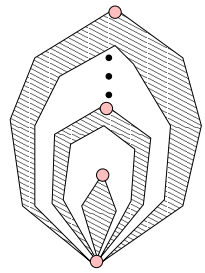


Figure 6: A class of embedded planar graphs whose optimal upward embeddings have $B + 1$ sources and sinks (circles).

tation of a (*weak-*)*visibility representation* of the input graph as a preliminary step [8]. In a visibility representation (see Figure 7), each vertex is mapped to a horizontal segment and each edge (u, v) is mapped to a vertical segment between the segments associated with u and v; horizontal segments do not overlap, and each vertical segment only intersect its extreme horizontal segments.

A standard technique [8] to compute a visibility representation of a plane graph G consists of calculating a bipolar orientation of G; if G is not biconnected it is augmented to a biconnected planar graph by adding a suitable number of dummy edges, which will be removed in the final drawing. However, this technique has several drawbacks: (i) Adding too many dummy edges may lead to a final drawing with area much bigger than necessary. On the other side, the problem of adding the minimum number of edges to make a planar graph biconnected and still planar is NP-hard [12]. (ii) Although an approximation algorithm for the above augmentation problem exists [9] (which reaches the optimal solution in many cases), implementing it efficiently is quite difficult, because it requires us to deal with the block tree of the graph and with an efficient incremental planarity testing algorithm. In fact, such an approximation algorithm has $O(n^2T)$ running time, where T is the amortized time bound per query or insertion operation of the incremental planarity testing algorithm. (iii) The presence of dummy edges in the graph makes difficult to deal with partial assignments of the upward embedding.

In [18] it is sketched a strategy for computing visibility representations of general connected graphs; such a strategy does not explicitly detail how to perform some necessary topological and geometric operations.

We propose the following algorithm for computing a visibility representation of a 1-connected embedded planar graph G_ϕ.

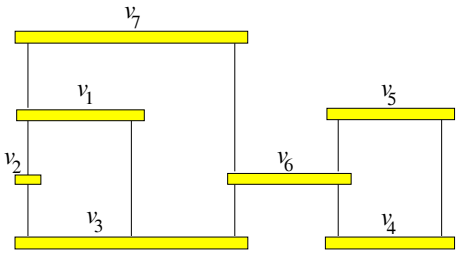


Figure 7: A visibility representation of the upward embedded graph shown in Figure 1(b).

Algorithm *Visibility-Upward-Embedding*

1. Compute an upward embedding $\mathcal{E}_\phi$ of G_ϕ, by calculating a feasible flow on network $\mathcal{N}_\phi$.

2. Compute an upward embedded st-graph S_ϕ including G_ϕ and preserving $\mathcal{E}_\phi$ on G_ϕ, by using the linear time saturation procedure described in [3] (note that, S_ϕ is biconnected).

3. Compute a visibility representation of S_ϕ (within its upward embedding) by using any known linear time algorithm [8], and then remove the edges introduced by the saturation procedure.

Algorithm *Visibility-Upward-Embedding* has $O(n \log n)$ running time, because its time complexity is dominated by the cost of computing a feasible flow on $\mathcal{N}_\phi$. We experimentally observed that the area of the visibility representations produced by this algorithm can be dramatically improved by computing upward embeddings with the minimum number of sources and sinks. To do that we just apply a min-cost-flow algorithm in Step 1. Clearly, in this case, the running time of the whole algorithm grows to $O(n^{\frac{7}{4}} \log n)$. The following theorem summarizes the main contributions of this section.

Theorem 3 *There exists an $O(n^{\frac{7}{4}} \log n)$ time algorithm that computes an upward embedding of an embedded 1-connected planar graph with the minimum number of sources and sinks.*

4.1 Experimentation

We present a preliminary study that shows how algorithm *Visibility Upward Embedding* can be slightly refined in order to get a certain control over the width and the height of visibility representations of 1-connected planar graphs. We start from the intuition that by re-arranging the blocks around the cutvertices in the upward embedding, it is possible to reduce the height or the width of the visibility representation. Namely, if v is a cutvertex and f a face which is doubly incident on v, placing all the blocks of v either above or below leads to a reduction of the height and to an increase in the width. Figure 8 gives an idea of this fact. Such re-arrangement is easily performed by exploiting the flow network associated with the plane graph. In such a network we can always move a unit of flow from an arc $a_1 = (v, f)$ to another arc $a_2 = (v, f)$ keeping the feasibility of the flow. Clearly, arcs a_1 and a_2 exist only if v is a cutvertex.

The experimentation has been performed on a randomly generated test suite of 1820 graphs whose number n of vertices ranges from 10 to 100 (20 instances for each value of n). The graphs are planar and equipped with an embedding. Each graph of the test suite has a number of cutvertices between $n/10$ and $n/5$, the number of blocks attached to a cutvertex is between 2 and 5, the number of cutvertices of a block is between 1 and 5, and each biconnected component is generated by using the algorithm in [2]. A detailed description of the procedure can be found in [16].

For each graph of the test suite, we first compute an upward embedding having the minimum number of sources and sinks that keeps the given embedding unchanged. We

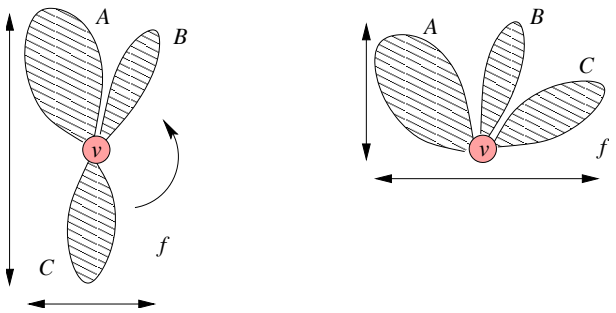
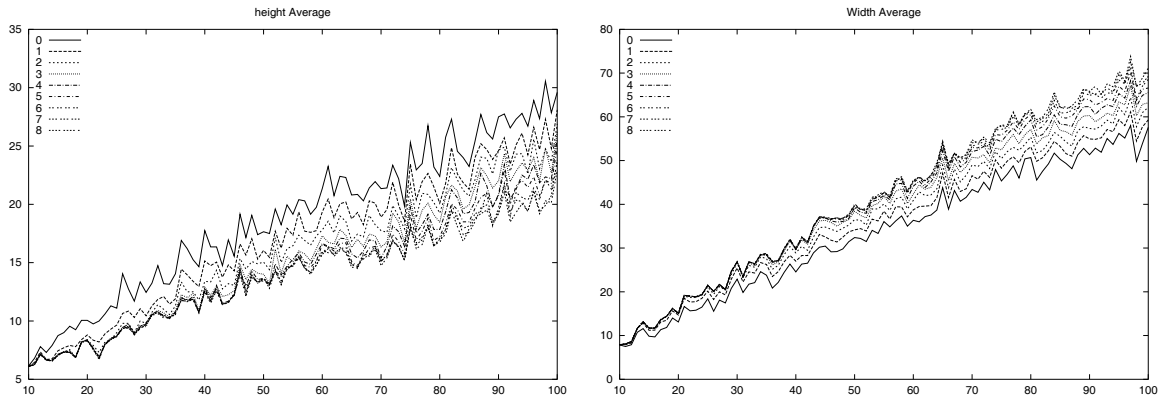


Figure 8: Placing all the blocks of a cutvertex above or below it allows a certain control on the height and the width of the visibility representation.



(a) height (b) width

Figure 9: The charts show how re-arranging the blocks around cutvertices affects the width and the height of the visibility representation.

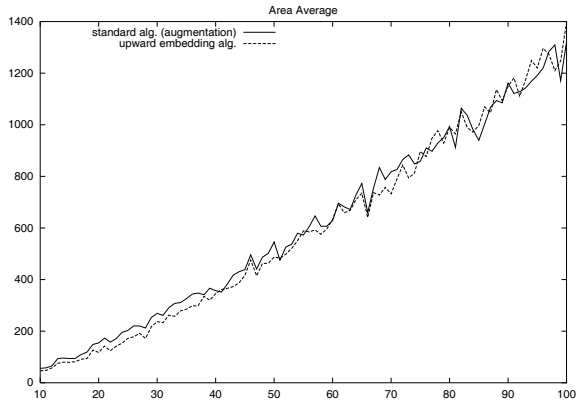


Figure 10: Area of the drawings computed with our strategy against the area of the drawings computed with a standard technique based on a sophisticated augmentation algorithm (average values). The x-axis represents the number of vertices.

proceed by redistributing the flow around at most a number k of cutvertices in such a way to place all their blocks above or below them. Note that, such a redistribution can be easily done in linear time. We average the width and the height on all the graphs having the same number of vertices. Charts in Figure 9 graphically show the results of the experimentation for k ranging from 0 to 8. It is interesting to observe how, for the same value of n and increasing k, the average of the width increases while the average of the height decreases.

Also, Figure 10 compares the area of the drawings computed with our strategy, where k is chosen equal to the total number of cutvertices of the graph, against the area of the drawings computed with a standard technique which uses the approximation algorithm in [9] to initially make the graph biconnected. In the two strategies we use the same algorithm for producing the visibility representation from the st-graph.

5 Open Problems

There are several open problems that we plan to study in the near future. For example, we are interested in an algorithm for counting and enumerating all upward embeddings of an embedded planar graph without repetitions. Also, is it possible to pass from an upward embedding to any other in linear time? Is there a linear time algorithm to compute optimal upward embeddings of plane graphs?

From an applications point of view we believe that the techniques shown in this paper may be successfully refined to compute drawings that approximate a given width/height ratio.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network flows. In G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors, *Optimization*, volume 1 of *Handbooks in Operations Research and Management*, pages 211–360. North-Holland, 1990.

[2] P. Bertolazzi, G. Di Battista, and W. Didimo. Computing orthogonal drawings with the minimum numbr of bends. *IEEE Transactions on Computers*, 49(8), 2000.

[3] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 6(12):476–497, 1994.

[4] P. Bertolazzi, G. Di Battista, C. Mannino, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM J. Comput.*, 27(1):132–169, 1998.

[5] M. Bousset. A flow model of low complexity for twisting a layout. In *Workshop of GD'93*, pages 43–44, Paris, 1993.

[6] J. Czyzowicz, A. Pelc, and I. Rival. Drawing orders with few slopes. Technical Report TR-87-12, Department of Computer Science, University of Ottawa, 1987.

[7] H. de Fraysseix, P. O. de Mendez, and P. Rosenstiehl. Bipolar orientations revisited. *Discrete Appl. Math.*, 56:157–179, 1995.

[8] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.

[9] S. Fialko and P. Mutzel. A new approximation algorithm for the planar augmentation problem. In *Symposium on Discrete Algorithms (SODA '98)*, pages 260–269, 1998.

[10] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes Comput. Sci.*, pages 286–297. Springer-Verlag, 1995.

[11] A. Garg and R. Tamassia. A new minimum cost flow algorithm with applications to graph drawing. In S. C. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 201–216. Springer-Verlag, 1997.

[12] G. Kant and H. L. Bodlaender. Planar graph augmentation problems. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes Comput. Sci.*, pages 286–298. Springer-Verlag, 1991.

[13] D. Kelly. Fundamentals of planar ordered sets. *Discrete Math.*, 63:197–216, 1987.

[14] D. Kelly and I. Rival. Planar lattices. *Canad. J. Math.*, 27(3):636–665, 1975.

[15] D. G. Kirkpatrick and S. K. Wismath. Weighted visibility graphs of bars and related flow problems. In *Proc. 1st Workshop Algorithms Data Struct.*, volume 382 of *Lecture Notes Comput. Sci.*, pages 325–334. Springer-Verlag, 1989.

[16] M. Pizzonia. *Engineering of Graph Drawing Algorithms for Applications.* PhD thesis, Dipartimento di Informatica e Sistemistica, Università "La Sapienza" di Roma, 2001.

[17] I. Rival. Reading, drawing, and order. In I. G. Rosenberg and G. Sabidussi, editors, *Algebras and Orders*, pages 359–404. Kluwer Academic Publishers, 1993.

[18] R. Tamassia and I. G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.*, 1(4):321–341, 1986.

[19] S. K. Wismath. *Bar-Representable Visibility Graphs and Related Flow Problems.* Ph.D. thesis, Dept. Comput. Sci., Univ. British Columbia, 1989.