ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
CONSIGLIO NAZIONALE DELLE RICERCHE

P. Bertolazzi, G. Di Battista, W. Didimo

QUASI-UPWARD PLANARITY

R. 499 Marzo 1999

Paola Bertolazzi – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email : bertola@iasi.rm.cnr.it.

Giuseppe Di Battista – Dipartimento di Informatica e Automazione, Università di Roma Tre, via della Vasca Navale 84, 00146 Roma, Italy. Email : gdb@dia.uniroma3.it.

Walter Didimo – Dipartimento di Informatica e Automazione, Università di Roma Tre, via della Vasca Navale 84, 00146 Roma, Italy. Email : didimo@dia.uniroma3.it.

Research supported in part by the ESPRIT LTR Project no. 20244 - ALCOM-IT.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica, CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.rm.cnr.it
URL: http://www.iasi.rm.cnr.it

Abstract

An upward planar drawing of a directed graph (digraph) is a planar drawing such that all the edges are represented by curves monotonically increasing in the vertical direction. In this paper we introduce and study the concept of quasi-upward planarity. Quasi-upward planarity allows to extend the upward planarity theory to a large class of digraphs including also digraphs that have directed cycles. First, we characterize the digraphs that have a quasi-upward planar drawing. Second, we give a polynomial time algorithm for computing "optimal" quasi-upward planar drawings within a given planar embedding. Further, we apply branch and bound techniques to the problem of computing optimal quasi-upward planar drawings, considering all possible planar embeddings. The paper contains also experimental results about the proposed techniques.

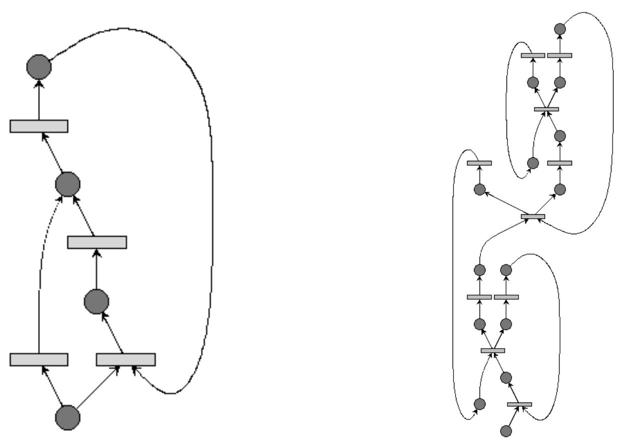
1. Introduction

An *upward drawing* of a digraph is a drawing such that all the edges are represented by curves monotonically increasing in the vertical direction. A digraph is *upward planar* if it has an upward planar drawing.

Upward planar drawings have been deeply investigated and several theoretical and application-oriented results can be cited in this intriguing field. What follows is a limited list containing a few examples (a survey on upward planarity can be found in [23]). Upward planarity of specific families of digraphs has been studied in: planar *st*-digraphs [36, 13, 16, 21], embedded and triconnected digraphs [3], single source digraphs [30, 29, 36, 27, 4], bipartite digraphs [12], outerplanar digraphs [35], trees [37, 8, 9, 20, 31, 38, 41], and hierarchical digraphs [32, 28]. The NP-completeness of the upward planarity testing is proved in [22]. Upward drawings on different surfaces are studied in [24, 33]. Orthogonal upward drawings are investigated in [18]. Further, an impressive set of results on upward drawings can be found in the literature on ordered sets.

Despite of such a long list of results, upward planar drawings have found, up to now, limited practical applicability. The reasons for this are mainly in the tightness of the upward planar standard that is satisfied by "a few" digraphs. Also, the applications require very often a similar but slightly different standard, where the drawing is upward "as much as possible". This is the case, for example, of Petri Nets [1] and of SADT diagrams [7].

In this paper we introduce and investigate quasi-upward planar drawings. A *quasi-upward drawing* of a digraph is such that the horizontal line through each vertex "locally" splits the incoming edges from the outgoing edges. The term locally is used to identify a sufficiently small connected region properly containing the vertex. Examples of quasi-upward (planar) drawings of Petri Nets are shown in Fig. 1.



(a) (b)

Figure 1: Two quasi-upward planar drawings of Petri Nets.

It is easy to see that an upward drawing is also a quasi-upward drawing and that, while an upward drawing requires the acyclicity of the digraph, any digraph can be drawn quasi-upward.

In a quasi-upward drawing we call *bend* a point on an edge where the tangent moves from the interval $0, \pi$ to the interval $\pi, 2\pi$ or viceversa. In other words, a bend is a point on an edge where the edge is tangent to a horizontal line. The quasi-upward drawings of Fig. 1.a and Fig. 1.b have 2 and 8 bends, respectively. An upward drawing is a quasi-upward drawing with 0 bends.

The main contributions of this paper are summarized as follows:

- We introduce the quasi-upward planar drawing convention (Section 3).
- We characterize the class of digraphs that have a quasi-upward planar drawing (Section 4).
- We give a polynomial time algorithm for computing a quasi-upward planar drawing with the minimum number of bends of a planar embedded digraph. We use a min-cost flow technique. This gives new insights on the techniques for orthogonal drawings presented in [39] and on those presented in [3] for upward planarity. We extend the technique to general (non-planar) digraphs. (Section 4).
- Motivated by the practical applicability of quasi-upward planar drawings, we study the problem of computing quasi-upward planar drawings with the minimum number of bends of planar digraphs considering all the possible planar embeddings. Thus, we present:
  - A modification of $SPQR$-trees [14, 15] that allows to decompose digraphs (Section 2).
  - Lower bounds techniques for quasi-upward planar drawings (Section 5.1).
  - A branch and bound algorithm for computing a quasi-upward planar drawing with the minimum number of bends of a biconnected digraph (Section 5.2). Such a technique is a variation of the technique presented in [5] for orthogonal drawings and can be used for each biconnected component of a digraph, constituting the basis of a powerful drawing heuristic. Further, the technique allows to test if the digraph is upward planar.
  - An implementation of the above branch and bound algorithm and the results of experiments performed on a test suite of 300 biconnected digraphs with number of vertices in the range $10--200$. The experiments show a reasonable time performance in the selected range. (Section 6).

2. Basic Notions and Preliminaries

We assume familiarity with planarity and connectivity of graphs [17, 34]. We also assume some familiarity with graph drawing [11].

2.1. Connectivity and Planarity

A separating k-set of a graph G is a set of k vertices whose removal increases the number of connected components of G. Separating 1-sets and 2-sets are called *cutvertices* and *separation pairs*, respectively. A connected graph is said to be *biconnected* if it has no cutvertices. Let G be a planar graph. An *embedded planar graph* G_ϕ is an equivalence class of planar drawings of G with the same ordering for the adjacency lists of vertices and with the same external face. Such a choice ϕ for an ordering of the adjacency lists and of an external face is called *planar embedding* of G.

A graph such that all the edges are directed is called *digraph*. An *st-digraph* is an acyclic digraph with exactly one source s and one sink t. The following definitions, usually introduced for graphs, are used here for digraphs.

Let G be a biconnected planar graph. A *split pair* of G is either a separation pair or a pair of adjacent vertices. A *maximal split component* of G with respect to a split pair $\{u, v\}$ (or simpler a maximal split component of $\{u, v\}$) is either an edge (u, v) or a maximal subgraph C of G such that C contains u and v, and $\{u, v\}$ is not a split pair of C. A vertex w distinct from u and v belongs to exactly one maximal split component of $\{u, v\}$.

Let $\{u, v\}$ be a split pair of G and let $\mathcal{C}_{uv}$ be the set of all maximal split components of $\{u, v\}$. We call *split component* of $\{u, v\}$ a subgraph G_{uv} of G that is the union of some elements of $\mathcal{C}_{uv}$.

Suppose $G_1, \ldots, G_k$ are some pairwise edge disjoint split components of G with respect to split pairs $u_1, v_1 \ldots u_k, v_k$, respectively. The graph G' obtained by substituting each of $G_1, \ldots, G_k$ with *virtual edges* $(u_1, v_1) \ldots (u_k, v_k)$ is a *partial graph* of G. We denote by E^{virt} ($E^{nonvirt}$) the set of virtual (non-virtual) edges of G'. We also say that G_i is the *pertinent graph* of (u_i, v_i) and that (u_i, v_i) is the *representative edge* of G_i.

In Fig. 2 it is shown an example of a partial digraph G' of a digraph G. G_1 and G_2 are two split components of G, and p_1 and p_2 are two virtual paths of G', corresponding to two paths of G_1 and G_2, respectively.

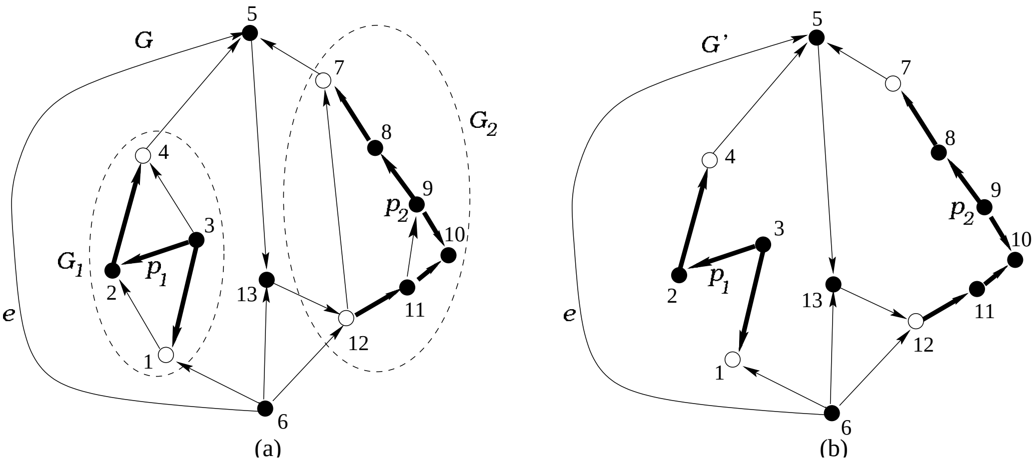


Figure 2: (a) A digraph G, and two split components G_1 and G_2. (b) A partial digraph of G, derived from G by replacing G_1 and G_2 with the virtual paths p_1 and p_2, respectively. Virtual paths are represented by thick edges.

Let ϕ be an embedding of G and ϕ' be an embedding of G'. We say that ϕ *preserves* ϕ' if $G'_{\phi'}$ can be obtained from G_ϕ by substituting each component G_i with its representative path.

2.2. Upward Planarity

An *upward drawing* of a digraph is such that all the edges are represented by curves monotonically increasing in the vertical direction. A digraph is *upward* if it admits an upward drawing.

Property 1. *A digraph is upward if and only if it is acyclic.*

An upward drawing that is also planar is an *upward planar drawing.* A digraph is *upward planar* if it admits an upward planar drawing.

Suppose that G_ϕ is an embedded digraph. A vertex v of G_ϕ is *bimodal* if the circular list of the edges around v can be partitioned into two (possibly empty) linear lists of edges, one consisting of the incoming edges and the other consisting of the outgoing edges. The embedding ϕ is *bimodal* if every vertex of G_ϕ is bimodal. A planar digraph is *bimodal* if it has a planar bimodal embedding.

Lemma 2.1. [3] *An embedded planar digraph G_ϕ has an upward planar drawing only if ϕ is bimodal.*

Now we briefly revise the results presented in [13, 29, 3, 11]. They give a characterization of the digraphs that are upward planar and describe a polynomial time algorithm for testing the existence of an upward planar drawing (and in case for computing it) of a digraph with a fixed planar embedding. Testing if a planar digraph G admits an upward planar drawing, overall the possible planar embeddings of G, has been shown to be an NP-complete problem [22]. We use the same terminology as [11].

The following theorem gives a general characterization of upward planarity.

Theorem 2.2. [13] *Let G be a digraph. The following statements are equivalent:*

1. *G is upward planar.*
2. *G admits an upward planar straight-line drawing.*
3. *G is the spanning subgraph of a planar st-digraph.*

In the rest of this subsection we consider planar digraphs with a fixed planar embedding. Let G_ϕ be an embedded planar digraph and let Γ be an upward planar drawing of G_ϕ. From Theorem 2.2, we can assume that Γ is a straight-line drawing.

The *angles* of G_ϕ are the pairs of consecutive edges incident on the same vertex v. Observe that an angle identifies a unique face f (the face containing its two edges) but in the case when v has degree 2. With the purpose to allow each angle to identify a unique face, we define an angle a as an ordered pair, where the ordering is given by visiting the faces adjacent to a counterclockwise. If a vertex has exactly one incident edge e, we call angle the pair e, e. The angles of G_ϕ are mapped to geometric angles in Γ. In particular, an angle e, e of G_ϕ corresponds to a 2π angle in Γ. An *internal vertex* of G is a vertex that is neither a source nor a sink. Let p any simple path in G. We call *source-switch* (*sink-switch*) of p, a vertex of p with a pair of incident outgoing (incoming) edges in p. If f is a face of G_ϕ, a *switch* of f is a switch of the border of f. The following properties hold.

Property 2. *For each face f of G_ϕ, the number of source-switches of f is equal to the number of sink-switches of f.*

Property 3. *Each source (sink) v of G is a source-switch (sink-switch) of every face of G_ϕ that contains v.*

Now, for each face f of G_ϕ, and for each angle a of f, we assign a label L to a, if a is larger than π in Γ. We denote with $L(f)$ the number of L labels on the angles of f, and with $A(f)$ the number of source-switches in f. Also, if v is a vertex of G, we denote with $L(v)$ the number of L labels on the angles at vertex v in Γ.

Lemma 2.3. [3] *For each vertex v and for each face f of G_ϕ, we have in Γ:*

$$L(v) = \begin{cases} 0 & \textit{if } v \textit{ is an internal vertex,} \\ 1 & \textit{if } v \textit{ is a source or a sink.} \end{cases}$$

$$L(f) = \begin{cases} A(f) - 1 & \textit{if } f \textit{ is an internal face,} \\ A(f) + 1 & \textit{if } f \textit{ is the external face.} \end{cases}$$

We call *capacity*(f) the number $A(f) - 1$ if f is an internal face, and the number $A(f) + 1$ if f is the external face. By using the bimodality property of ϕ and the Euler's theorem for planar graphs, the following lemma can be proved. It is a simple variation of a lemma proved in [3].

Lemma 2.4. *The total number of source and sink vertices of G is equal to the sum of capacity(f) for each face f of G_ϕ.*

Motivated by Lemma 2.3 and by Lemma 2.4, we now consider an assignment of L labels to angles at source and sink vertices of G_ϕ. We say that this assignment is *upward consistent* if:

(a). For each source or sink v, exactly one angle a at v is labeled L. We say that v is *assigned* to the face identified by a.

(b). For each face f, the number of L labels on angles at sinks and sources of f is equal to *capacity*(f).

The following result [3] gives a characterization of the upward planarity of embedded digraphs.

Theorem 2.5. [3] *Let G_ϕ be an embedded planar digraph. There exists an upward planar drawing of G if and only if the following properties hold:*

- *G is acyclic;*
- *ϕ is bimodal;*
- *G_ϕ admits an upward consistent assignment.*

Observe that an upward consistent assignment of G_ϕ corresponds to an equivalence class of upward planar drawings of G_ϕ. In all the drawings of the class, for each vertex v that is assigned to a face f, there is an angle larger than π at vertex v in face f. We also remind that, once an upward consistent assignment is computed, it is possible to construct from it an upward planar drawing of the corresponding equivalence class, with a linear time algorithm [3, 11]. This is done by determining a planar st-digraph that includes G_ϕ, by computing an upward planar drawing of the st-digraph, and then removing from the drawing the dummy edges added during the first step.

2.3. $SPQR$-trees for Digraphs

In the following we revise $SPQR$-trees, presented in [14, 15] as a decomposition technique for graphs, with the purpose to use them to decompose digraphs instead of graphs. $SPQR$-trees are closely related to the classical decomposition of biconnected graphs into triconnected components [25].

Let $\{s,t\}$ be a split pair of G. A *maximal split pair* $\{u,v\}$ of G with respect to $\{s,t\}$ is a split pair of G distinct from $\{s,t\}$ such that for any other split pair $\{u',v'\}$ of G, there exists a split component of $\{u',v'\}$ containing vertices u, v, s, and t.

Let $e=(s,t)$ be an edge of G, called *reference edge*. The *$SPQR$-tree* $\mathcal{T}$ of G with respect to e describes a recursive decomposition of G induced by its split pairs. Tree $\mathcal{T}$ is a rooted ordered tree whose nodes are of four types: S, P, Q, and R. Each node μ of $\mathcal{T}$ has an associated biconnected multigraph containing directed and undirected edges, called the *skeleton* of μ, and denoted by $skeleton(\mu)$. Also, it is associated with an edge of the skeleton of the parent ν of μ, called the *virtual edge* of μ in $skeleton(\nu)$. Tree $\mathcal{T}$ is recursively defined as follows.

If G consists of exactly two parallel edges between s and t, then $\mathcal{T}$ consists of a single Q-node whose skeleton is G itself (trivial case). If the split pair $\{s,t\}$ has at least three split components $G_1,\cdots,G_k$ ($k\geq 3$), the root of $\mathcal{T}$ is a P-node μ. Graph $skeleton(\mu)$ consists of k parallel undirected edges between s and t, denoted $e_1,\cdots,e_k$, with $e_1=e$. Otherwise, the split pair $\{s,t\}$ has exactly two split components, one of them is the reference edge e, and we denote with G' the other split component. If G' has cutvertices $c_1,\cdots,c_{k-1}$ ($k\geq 2$) that partition G into its blocks $G_1,\cdots,G_k$, in this order from s to t, the root of $\mathcal{T}$ is an S-node μ. Graph $skeleton(\mu)$ is the cycle of undirected edges $e_0,e_1,\cdots,e_k$, where $e_0=e$, $c_0=s$, $c_k=t$, and e_i connects c_{i-1} with c_i ($i=1\cdots k$).

If none of the above cases applies, let $\{s_1,t_1\},\cdots,\{s_k,t_k\}$ be the maximal split pairs of G with respect to $\{s,t\}$ ($k\geq 1$), and for $i=1,\cdots,k$, let G_i be the union of all the split components of $\{s_i,t_i\}$ but the one containing the reference edge e. The root of $\mathcal{T}$ is an R-node μ. Graph $skeleton(\mu)$ is obtained from G by replacing each subgraph G_i with the undirected edge e_i between s_i and t_i.

Except for the trivial case, μ has children $\mu_1,\cdots,\mu_k$ in this order, such that μ_i is the root of the $SPQR$-tree of graph $G_i\cup e_i$ with respect to reference edge e_i ($i=1,\cdots,k$). Edge e_i is said to be the *virtual edge* of node μ_i in $skeleton(\mu)$ and of node μ in $skeleton(\mu_i)$. Digraph G_i is called the *pertinent digraph* of node μ_i, and of edge e_i. Finally, the extremal vertices of the reference edge of $skeleton(\mu)$ are called *poles* of $skeleton(\mu)$.

The tree $\mathcal{T}$ so obtained has a Q-node associated with each edge of G, except the reference edge e. We complete the $SPQR$-tree by adding another Q-node, representing the reference edge e, and making it the parent of the root node, so that it becomes the root.

Let μ be a node of $\mathcal{T}$. We have: if μ is an R-node, then $skeleton(\mu)$ is a triconnected graph; if μ is an S-node, then $skeleton(\mu)$ is a cycle; if μ is a P-node, then $skeleton(\mu)$ is a triconnected multigraph consisting of a bundle of multiple edges; and if μ is a Q-node, then $skeleton(\mu)$ is a biconnected multigraph consisting of two multiple edges.

The $SPQR$-trees of G with respect to different reference edges are isomorphic and are obtained one from the other by selecting a different Q-node as the root. Hence, we can define the *unrooted $SPQR$-tree* of G without ambiguity.

The $SPQR$-tree $\mathcal{T}$ of a digraph G with n vertices and m edges has m Q-nodes and $O(n)$ S-, P-, and R-nodes. Also, the total number of vertices of the skeletons stored at the nodes of $\mathcal{T}$ is $O(n)$.

By applying the above definitions, the skeletons of the nodes of $\mathcal{T}$ contain both directed and undirected edges. To avoid this, we modify $\mathcal{T}$ as follows. For each node μ, each virtual edge (u, v) of $skeleton(\mu)$, different from the reference edge, is replaced by any simple path (*virtual path*) of the pertinent digraph of (u, v) between u and v. The reference edge (s, t) of $skeleton(\mu)$ is replaced with a simple path between s and t that has no edges in the pertinent digraph of node μ. We call *reference path* the path that substitutes the reference edge. Observe that the reference edge of $\mathcal{T}$ is the reference path of the skeleton of the child of the root node. After this modification, each skeleton is a subgraph of G.

In Fig. 3 it is shown the $SPQR$-tree of digraph G in Fig. 2, with respect to the reference edge e. It is equipped with the skeletons of some nodes.

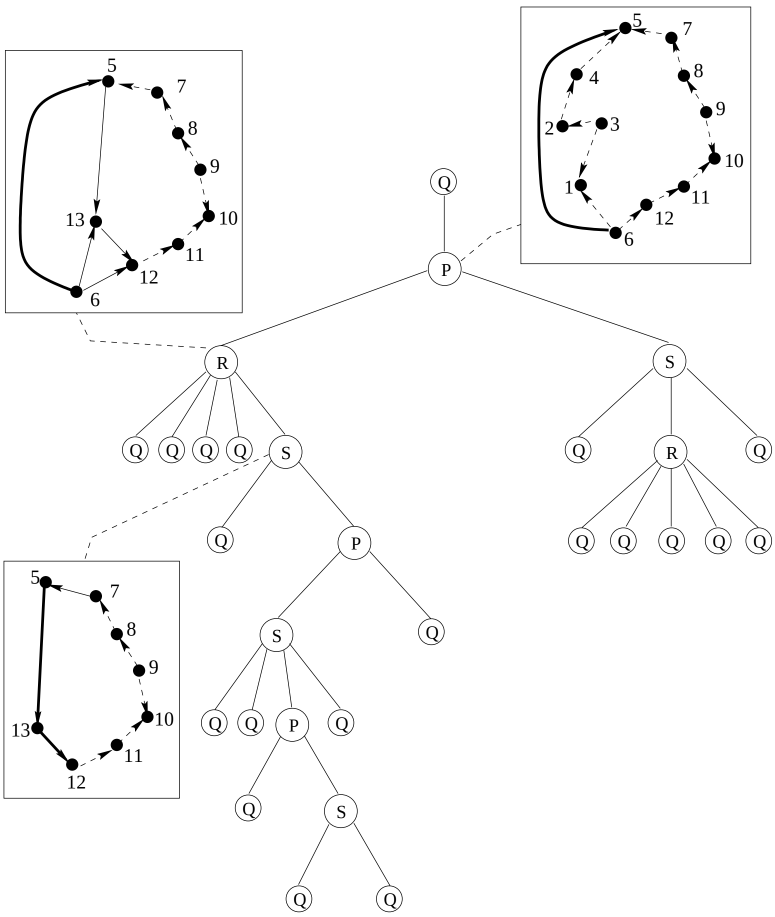


Figure 3: $SPQR$-tree of the digraph G in Fig. 2, equipped with the skeletons of some nodes. The virtual paths of the skeletons are dashed, and the edges in the reference paths are thick.

A digraph G is planar if and only if the skeletons of all the nodes of the $SPQR$-tree $\mathcal{T}$ of G are planar. An $SPQR$-tree $\mathcal{T}$ rooted at a given Q-node represents all the planar embeddings of G having the reference edge (associated with the Q-node at the root) on the external face. These embeddings can be obtained by combining all the different planar embeddings of the skeletons of P- and R-nodes of $\mathcal{T}$. (see Section 5).

3. Quasi-Upward Planarity

As shown in Section 2 the acyclicity and the bimodality are only necessary conditions for the existence of an upward planar drawing of a planar digraph, but they are not sufficient. For example, Fig. 4.(a) shows a planar digraph that is acyclic and bimodal, but that is not upward planar.

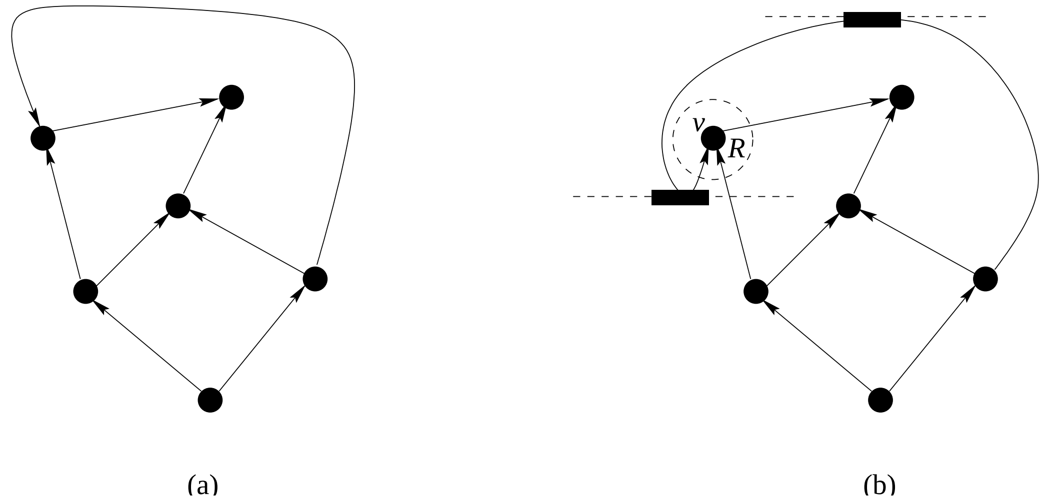


Figure 4: (a) An example of digraph that is not upward planar. (b) A quasi-upward planar drawing of the same digraph, with two bends.

Actually, there are many practical applications where the graphs to be drawn are not upward planar. In these cases it is useful to find a drawing of the digraph that is upward "as much as possible". This motivates the following definition.

Let G be a digraph. A drawing Γ of G is *quasi-upward* if for each vertex v there exists a sufficiently small circular connected region R of the plane, properly containing v, such that, in the intersection of R with Γ, the horizontal line through v separates the incoming edges (below the line) from the outgoing edges (above the line). See Fig. 4.(b).

For the sake of simplicity, and without loss of generality, we restrict the attention to drawings that have, along the edges, a finite number of points that are tangent to a horizontal line. It would be easy to remove this assumption by considering each connected infinite set of tangent points as a single point.

In a quasi-upward drawing we call *bend* a point q on an edge (u, v), distinct from u and v, such that (u, v) is tangent to the horizontal line through q (see Fig. 4.(b)). We also say that q is a *right bend* (*left bend*) if the edge (u, v) turns to right (left) at q, while walking on (u, v) from u to v.

A digraph G is *quasi-upward* if it admits a quasi-upward drawing. It is easy to see that:

Property 4. *All digraphs are quasi-upward.*

Let G be a planar digraph. A quasi-upward drawing of G that is also planar is a *quasi-upward planar drawing* of G. Digraph G is *quasi-upward planar* if it admits a quasi-upward planar drawing. Fig. 4.(b) shows a quasi-upward planar drawing of the digraph in Fig. 4.(a). The two horizontal dashed lines are the tangents at the bend points. The bends are put in evidence by small circles.

Property 5. *An upward drawing is a quasi-upward drawing.*

Property 6. *A quasi-upward drawing without bends is an upward drawing.*

The following lemma can be easily proved by geometric considerations.

Lemma 3.1. *In a quasi-upward drawing each edge has an even number of bends.*

We now introduce an operation on an embedded planar digraph G_ϕ.

Let (u, v) be an edge of G_ϕ, and let $k > 0$ be an integer number. A $2k$*-insert-switch* operation on edge (u, v) removes (u, v) and inserts vertices $v_1, u_1, v_2, u_2, \dots, v_k, u_k$ and directed edges $(u, v_1), (u_1, v_1), (u_1, v_2), (u_2, v_2), \dots, (u_k, v)$ (See Fig. 5). The new path is planarly embedded in place of (u, v). Observe that such a path has $2k$ switches: $u_1, \dots, u_k$ are k new sources, and $v_1, \dots, v_k$ are k new sinks of G_ϕ.

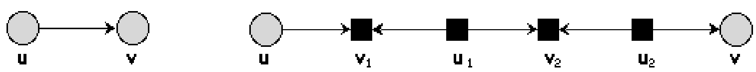


Figure 5: An example of 4-insert-switch. The 4 switches are represented by squares.

Let $G'_{\phi'}$ be an embedded digraph obtained from G_ϕ by applying the $2k$-insert-switch operation on some edges of G_ϕ. We say that $G'_{\phi'}$ is *switch-morphic* to G_ϕ. Note that if the direction of the edges is not considered, then a digraph that is switch-morphic to another digraph is also homeomorphic to that digraph.

Suppose that Γ is a quasi-upward planar drawing of G_ϕ. We construct an upward planar drawing Γ' from Γ as follows. Let (u, v) be an edge with $2k$ bends (see Lemma 3.1), with $k > 0$. We replace each bend of (u, v) with a vertex and orient upward all the edges that result from this replacement. We apply the same update to all the edges of Γ that have bends. We have that Γ' is an upward planar drawing of a digraph $G'_{\phi'}$ that is switch-morphic to G_ϕ. We say that Γ' is the *underlying upward planar drawing* of Γ. Fig. 6 shows an example of underlying upward planar drawing of a quasi-upward planar drawing with 6 bends.

We observe that each upward planar drawing Γ' of $G'_{\phi'}$ is the underlying upward planar drawing of a quasi-upward planar drawing Γ of G_ϕ. Namely, Γ can be derived from Γ' as follows. For each path $p = (u, v_1, u_1, v_2, \dots, u_k, v)$ of $G'_{\phi'}$ that results from a $2k$-insert-switch operation on an edge (u, v) of G_ϕ:

- make all the edges of p undirected, except for (u_k, v);
- replace all vertices u_i, v_i, $(i = 1, \dots, k)$ with bend points.

In the set of the quasi-upward drawings of a digraph G, it is useful to distinguish a specific subset. Let Γ be a quasi-upward drawing of G. We say that Γ is a *proper quasi-upward drawing*

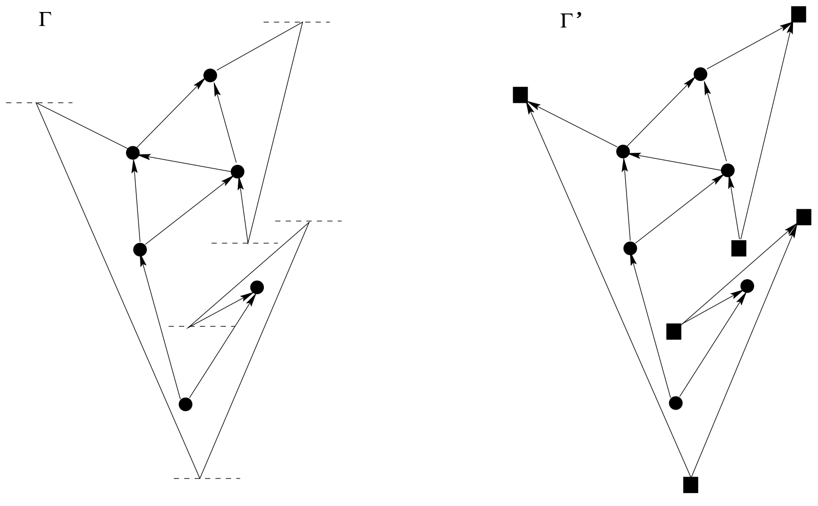


Figure 6: (a) A quasi-upward planar drawing with 6 bends and (b) its underlying upward planar drawing. The little squares represent the vertices replacing bends

if each edge of Γ has an even number of right bends. We observe that, by Lemma 3.1, if Γ is proper it also has on each edge an even number of left bends. For example, the quasi-upward planar drawings of Fig. 4.(b) and Fig. 6.(a) are proper.

Lemma 3.2. *For each quasi-upward planar drawing Γ of a planar embedded digraph G_ϕ that is not proper, there exists a proper quasi-upward planar drawing of G_ϕ that has less bends than Γ and that, for each edge, has either only left bends or only right bends.*

Proof. Let (u, v) be an edge of Γ that has $l + r = 2k$ bends, where l (r) is odd and is equal to the number of left (right) bends. We assume, without loss of generality, $0 < l \leq r$.

Consider the underlying upward planar drawing Γ' of Γ. Consider the upward planar embedded digraph $G'_{\phi'}$ that is switch-morphic to G_ϕ and that admits Γ' as an upward planar drawing. Let p be the path replacing (u, v) in $G'_{\phi'}$. Path p has $2k$ vertices (not considering the end-vertices u and v).

Let f_l and f_r be the two (not necessarily distinct) faces to the left and to the right of p while walking on p from u to v.

Consider the upward consistent assignment of $G'_{\phi'}$ describing the equivalence class of Γ'. We have that the vertices of p that replace the left bends are assigned to f_r. In fact, they form an angle greater than π in f_r. Analogously, the vertices of p that replace the right bends are assigned to f_l.

We can express the capacity of f_l as $capacity(f_l) = k + s_{f_l}$, where k is the contribution to the capacity given by the k source-switches along p and s_{f_l} is the rest of the capacity. Analogously, $capacity(f_r) = k + s_{f_r}$. Also, in the upward consistent assignment we have $k + s_{f_l} = r + a_{f_l}$, where r is the number of sources and sinks of p assigned to f_l and a_{f_l} is the rest of the sources and sinks assigned to f_l. Analogously, we can write $k + s_{f_r} = l + a_{f_r}$. Hence, $a_{f_l} = k + s_{f_l} - r$ and $a_{f_r} = k + s_{f_r} - l$.

We susbstitute in $G'_{\phi'}$ path p with a new path p' consisting of $2k-2l$ switches. Observe that $2k-2l>0$ since $2k-2l=r+l-2l=r-l$ and $r>l$. We call $G''_{\phi''}$ the resulting planar embedded digraph. Now, we show that $G''_{\phi''}$ admits an upward consistent assignment.

We assign all sources and sinks of $G''_{\phi''}$ that are not on p' to the same faces they were assigned in $G'_{\phi'}$. We assign all the $r-l$ sources and sinks of p' to f_l.

After the replacement of p with p' the capacity of f_l becomes $capacity'(f_l)=k-l+s_{f_l}$. The number of sources and sinks assigned to f_l becomes $r-l+a_{f_l}$. Substituting the expression of a_{f_l} with $k+s_{f_l}-r$ we have that the capacity and the number of sources and sinks assigned to f_l coincide. On the other hand, $capacity'(f_r)=k-l+s_{f_r}$, while the number of sources and sinks assigned to f_r becomes a_{f_r}. Substituting a_{f_r} with $k+s_{f_r}-l$ we have that the capacity and the number of sources and sinks assigned to f_r coincide.

We conclude that $G''_{\phi''}$ has an upward consistent assignment. Thus, it has an upward planar drawing. Such a drawing is the underlying upward planar drawing of a quasi upward planar drawing of G_ϕ with less bends than Γ and with only right bends on edge (u,v).

The proof is completed by applying the same reasoning to all the edges of Γ that have both right bends and left bends. ∎

4. Computing Quasi-Upward Planar Drawings with the Minimum Number of Bends

In this section first we give a characterization of quasi-upward planarity. Second, if an embedded digraph is quasi-upward planar, we give a polynomial time algorithm for computing for it a quasi-upward planar drawing with the minimum number of bends. Further, we show a method for constructing quasi-upward drawings of general digraphs.

We start by proving the following theorem.

Theorem 4.1. *Let G be a digraph. There exists a quasi-upward planar drawing of G if and only if G is planar bimodal.*

We remark that the characterization of Theorem 4.1 allows to assert that a quasi-upward planar drawing of a bimodal digraph exists even if the digraph is not acyclic.

First of all, we observe that if ϕ is not bimodal, it is obvious that a quasi-upward planar drawing of G_ϕ does not exist. Then, we have just to prove that the bimodality of ϕ is a sufficient condition for the existence of a quasi-upward planar drawing of G_ϕ.

To do that, we define a flow network $\mathcal{N}$, associated with G_ϕ. Then we prove that there always exists an integer feasible flow in $\mathcal{N}$ that corresponds to an equivalence class of proper quasi-upward planar drawings of G_ϕ. We also prove that, for each proper quasi-upward planar drawing of G_ϕ there is an associated integer flow in the network $\mathcal{N}$. Further, we show that the total cost of the flow associated with a quasi-upward planar drawing Γ of G_ϕ is equal to the total number of bends of Γ.

Let G_ϕ be an embedded bimodal planar digraph. In the following we describe the flow network $\mathcal{N}$ associated with G_ϕ.

For each arc e of $\mathcal{N}$ we denote by $\beta(e)$ and $\chi(e)$ the capacity and the cost of e, respectively. Also, we denote by $\sigma(e)$ the flow on edge e.

- Nodes of $\mathcal{N}$ are all the sources and sinks (*vertex-nodes*), and all the faces (*face-nodes*) of G_ϕ;

- For each source or sink v that belongs to a face f, there is an arc (v, f) in $\mathcal{N}$ such that: $\beta(v, f) = 1$ and $\chi(v, f) = 0$;
- For each pair of adjacent faces f and g, there is a pair of arcs (f, g) and (g, f) in $\mathcal{N}$ such that: $\beta(f, g) = \beta(g, f) = +\infty$ and $\chi(f, g) = \chi(g, f) = 2$;
- Each vertex-node of $\mathcal{N}$ supplies a flow equal to 1;
- Each face-node of $\mathcal{N}$ that corresponds to a face f that is not a directed cycle demands a flow equal to *capacity*(f). If f is a directed cycle then *capacity*$(f) = -1$; in this case f supplies a flow equal to 1.

The following result is an immediate consequence of Lemma 2.4 and of the fact that all the arcs of $\mathcal{N}$ between face-nodes have an infinite capacity.

Lemma 4.2. *There always exists an integer feasible flow in the network $\mathcal{N}$.*

The relationship between the network $\mathcal{N}$ and quasi-upward planar drawings is put in evidence in the following theorem.

Theorem 4.3. *Let G_ϕ be an embedded bimodal planar digraph, and let $\mathcal{N}$ be the flow network associated with G_ϕ.*

1. *For each integer feasible flow σ in $\mathcal{N}$ there exists a proper quasi-upward planar drawing Γ of G_ϕ such that the total number of bends of Γ is equal to the cost of σ.*
2. *For each proper quasi-upward planar drawing Γ of G_ϕ there exists an integer feasible flow σ in $\mathcal{N}$ such that the cost of σ is equal to the number of bends of Γ.*

Proof. Let σ be an integer feasible flow in $\mathcal{N}$. We show that there is a proper quasi-upward planar drawing Γ associated with σ, and that the number of bends of Γ is equal to the total cost of σ. To do that, we prove that it is possible to derive from σ an embedded planar digraph $G'_{\phi'}$ that is switch-morphic to G_ϕ and that is upward planar. Thus, $G'_{\phi'}$ has an upward planar drawing Γ'. We have that Γ' is an underlying upward planar drawing of a quasi-upward planar drawing Γ of G_ϕ.

Embedded planar digraph $G'_{\phi'}$ and an upward consistent assignment for $G'_{\phi'}$ are constructed as follows.

1. For each pair of adjacent faces f and g such that $\sigma(f, g) = k > 0$, execute a $2k$-insert-switch operation on an exactly one edge e shared by f and g (any edge with this property is suitable). Further, assign all the new $(2k)$ sources and sinks to g.
2. For each source or sink v incident on a face f such that $\sigma(v, f) = 1$, assign v to f.

The above assignment is upward consistent. In fact, consider any face f of G_ϕ and let f' be the face of $G'_{\phi'}$ obtained from the face f of G_ϕ by applying the above transformations. Denote by $Adj(f)$ the set of faces adjacent to f and by $S(v)$ the set of sources and sinks incident to f.

From the consistency of σ we have that:

$$\mathit{capacity}(f) = \sum_{g \in Adj(f)} \sigma(g, f) + \sum_{v \in S(f)} \sigma(v, f) - \sum_{g \in Adj(f)} \sigma(f, g).$$

Since each unit of flow exchanged between two adjacent faces causes the insertion of one source and one sink on the border of the two faces, the capacity of f' can be expressed by:

$$capacity(f') = capacity(f) + \sum_{g \in Adj(f)} \sigma(g,f) + \sum_{g \in Adj(f)} \sigma(f,g).$$

We can replace $capacity(f)$ obtaining:

$$capacity(f') = 2 \sum_{g \in Adj(f)} \sigma(g,f) + \sum_{v \in S(f)} \sigma(v,f).$$

At this point the upward consistency of the assignment is easily verified by observing that the number of sources and sinks assigned to f' is equal to $2\sum_g \sigma(g,f) + \sum_v \sigma(v,f)$.

Since $G'_{\phi'}$ is switch-morphic to G_ϕ and is upward planar it has an upward planar drawing that is an underlying upward planar drawing of a quasi-upward drawing Γ of G_ϕ. Further, since the flow $\sigma(g,f)$ has cost $2\sigma(g,f)$ and causes the insertion of $2\sigma(g,f)$ sources and sinks that correspond to bends of Γ, the total number of bends of Γ is equal to the cost of σ. Also, Γ is proper.

Conversely, let Γ be a proper quasi-upward planar drawing of G_ϕ. We have to prove that there exists a feasible flow σ in $\mathcal{N}$, such that the cost of σ is equal to the total number of bends of Γ. Let Γ' be the underlying upward planar drawing of Γ and let f be a face of Γ. We denote by $b_r(f)$ the total number of right bends on the edges of f when walking on the border of f clockwise. Similarly, we denote by $b_l(f)$ the number of left bends on the edges of f when walking on the border of f clockwise. Also, we denote by $V(f)$ the number of sources and sinks of Γ that have an angle larger than π in f.

Each left bend in a face f of Γ corresponds to a source or a sink with an angle larger than π in f', where f' is the face of Γ' that corresponds to the face f of Γ. Analoguosly, each right bend in f corresponds to a source or a sink with an angle less or equal than π in f'. Thus, we have:

$$capacity(f') = capacity(f) + (b_r(f) + b_l(f))/2.$$

Further, because Γ' is an upward planar drawing, from the properties of the upward consistent assignment we have that:

$$capacity(f') = V(f) + b_l(f).$$

From the two equalities above, we obtain that:

$$capacity(f) = V(f) + b_l(f)/2 - b_r(f)/2.$$

We now construct from Γ the flow σ as follows:

1. We start by setting a zero flow on each arc of $\mathcal{N}$.
2. For each source or sink v that has an angle larger than π in a face f of Γ, we set $\sigma(v,f) = 1$.
3. For each edge (u,v) of Γ, let f and g (not necessarily distinct) be the right face and left face of (u,v) while walking on (u,v) from u, and let b_r and b_l be the number of right bends and left bends of (u,v) while walking on (u,v) from u. We increase $\sigma(f,g)$ of $b_r/2$ units and $\sigma(g,f)$ of $b_l/2$ units. Observe that both $b_r/2$ and $b_l/2$ are integer values, because Γ is proper.

Since there is exactly one angle larger than π at each source and sink v of Γ, for each face f incident to v, the flow $\sigma(v, f)$ in $\mathcal{N}$ does not exceed the capacity of the arc (v, f), and $\sum_{f:v \in f} \sigma(v, f) = 1$, that is equal to the flow supplied by v.

Further, from the construction of σ, for each face f of Γ, we have that the total flow of N entering in f is equal to $V(f) + b_l(f)/2$ and the total flow leaving f is equal to $b_r(f)/2$. Then, by using the above equalities, we have that the flow through f in $\mathcal{N}$ is equal to the *capacity*(f), that is the flow demanded (or supplied) by the face f.

Let F be the set of faces of G_ϕ. To complete the proof, we observe that the total cost of the flow σ is as follows:

$$cost(\sigma) = \sum_{f \in F} (\sum_{g \in Adj(f)} \sigma(f, g)) = \sum_{f \in F} b_r(f),$$

that is the total number of bends of Γ.

∎

Lemma 4.2 and Theorem 4.3 allow to assert that, if ϕ is a planar bimodal embedding, there always exists a quasi-upward planar drawing of G_ϕ. Then, we have proved Theorem 4.1.

Further, Theorem 4.3 and Lemma 3.2 allow to reduce the problem of finding a quasi-upward planar drawing of G_ϕ, with the minimum number of bends (within ϕ), to a minimum cost flow problem in the network $\mathcal{N}$.

We summarize the above discussion by showing a schematic description of an algorithm that computes a quasi-upward planar drawing of an embedded bimodal digraph G_ϕ, with the minimum number of bends.

Algorithm 1. *Min-Bends-Quasi-Upward-Planar-Drawing*

- *Input :* An embedded bimodal digraph G_ϕ.
- *Output:* A quasi-upward planar drawing Γ of G_ϕ, with the minimum number of bends.

1. Build the flow network $\mathcal{N}$ associated with G_ϕ.
2. Compute a minimum cost flow σ in $\mathcal{N}$.
3. Construct from σ an upward planar drawing Γ' (that is an underlying upward drawing of Γ).
4. Construct from Γ' a quasi-upward planar drawing Γ, by replacing each dummy source and sink with a bend point.

We now analyze the complexity of Algorithm 1. We denote by n the number of vertices of G_ϕ. Step 1 can be easily executed in time $O(n)$. Step 2 can be executed in time $O(n^2 \log n)$, by using standard flow techniques (see e.g. [2]). Step 3 is executed in time $O(n + b)$ by using one of the algorithms described in [11], where b is the total cost of the flow σ (that is equal to the number of dummy sources and sinks of Γ').

We observe that $b = O(n^2)$. In fact, the value $|\sigma|$ of the flow σ is equal to the total number of sources and sinks of G_ϕ; also, since the flow has minimum cost, the cost of each unit of flow in the network is at most $2|F|$, where F is the set of all faces of G_ϕ.

The following theorem summarizes the time-complexity analysis.

Theorem 4.4. *Let G_ϕ be an embedded bimodal planar digraph. Algorithm 1 constructs in $O(n^2 logn)$ time a quasi-upward planar drawing of G_ϕ with the minimum number of bends.*

We observe that, in the final drawing computed by Algorithm 1, we can use Bezier curves (instead of polygonal lines) to represent edges, in order to aesthetically improve the drawing.

Fig. 7 is an illustration of the steps of Algorithm 1 on a simple input digraph with a bimodal planar embedding (see Fig. 4.(a)).

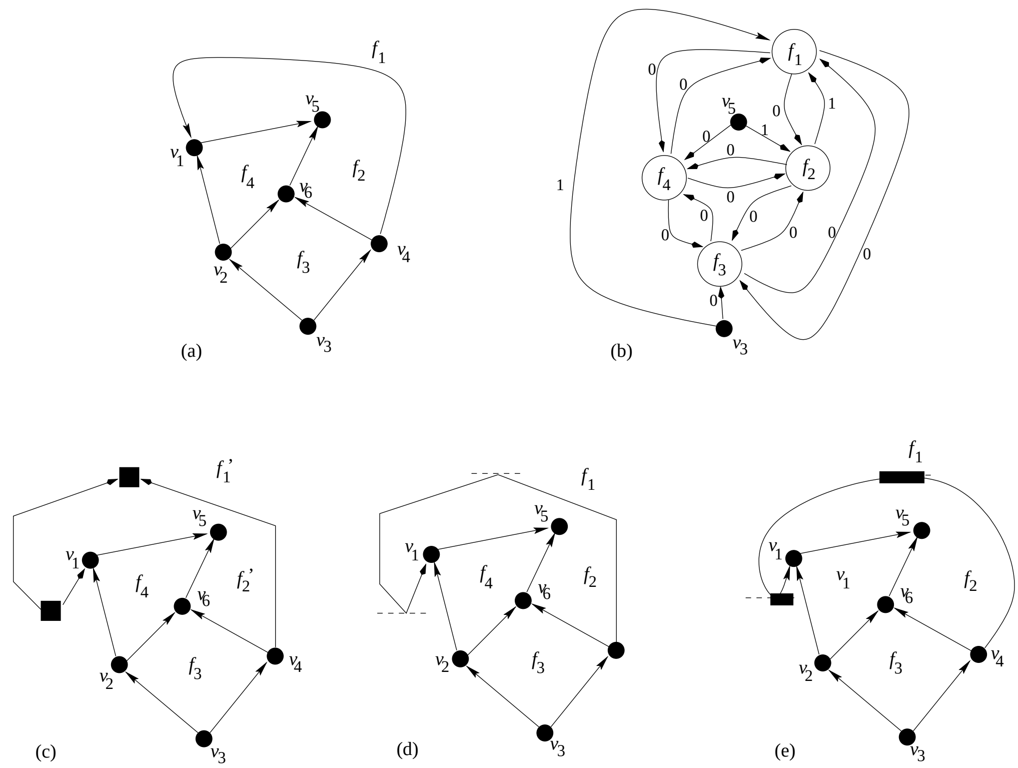


Figure 7: Illustration of Algorithm 1. (a) The input embedded bimodal digraph G_ϕ. (b) Network $\mathcal{N}$ associated with G_ϕ (Step 1) and its min-cost-flow σ (Step 2). Each edge is labelled with its flow. The total cost of the flow is 2. (c) Undelying planar upward drawing Γ' (Step 3). (d) The final quasi-upward planar drawing Γ obtained from Γ' by replacing dummy sources and sinks (squares) with bend points (Step 4). Γ has two bends. (e) A drawing obtained from Γ by using smoothed edges instead of polygonal-lines.

A further example of a quasi-upward planar drawing computed by an implementation of Algorithm 1 is given in Fig. 8.

We conclude this section by providing a polynomial time algorithm to find a quasi-upward drawing of a general directed graph G (that is not necessarily planar bimodal). Further, this algorithm finds a quasi-upward planar drawing of G if there exists one. The algorithm is the following:

Algorithm 2. *Quasi-Upward-Drawing*

- *Input :* A digraph G.

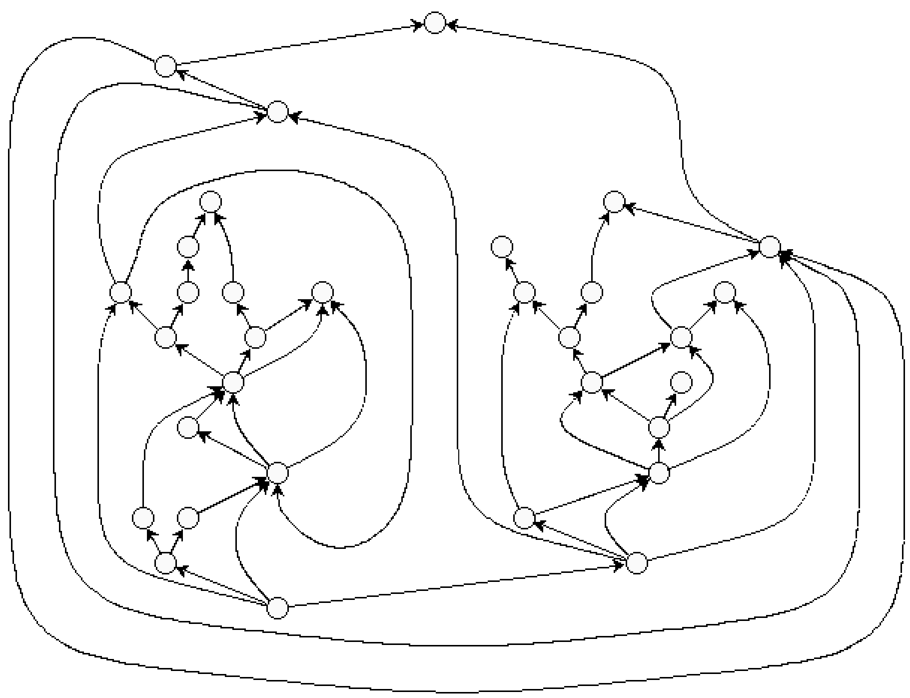


Figure 8: A quasi-upward planar drawing computed by an implementation of Algorithm 1.

- *Output:* A quasi-upward drawing Γ of G. Γ is planar if G is planar bimodal.

1. For each vertex v of G with at least two incoming edges and two outgoing edges, split v into two vertices v_1 and v_2, v_1 being incident to the incoming edges of v, and v_2 being incident to the outgoing edges of v; insert a dummy edge directed from v_1 to v_2. Denote by G' the digraph so obtained.

3. Execute a planarity test on G'. If there is a planar embedding of G', then go to Step 5. (The digraph is planar bimodal.)

4. Apply a planarization algorithm to G' with the constraint that no crossings must be present on the dummy edges introduced by Step 1.

5. For each dummy edge (v_1, v_2) introduced by Step 1, contract (v_1, v_2) into a vertex v, preserving the ordering of the incoming and outgoing edges incident on v_1 and v_2, respectively. Denote by G''_ϕ the planar embedded bimodal digraph so obtained.

6. Apply Algorithm 1 to G''_ϕ to dermine a quasi-upward planar drawing of G''_ϕ.

7. Remove, in the drawing, all dummy vertices introduced during Step 4 (if any), by replacing them with crossings.

We now explain in more details the above steps, and we show the correctness of Algorithm 2. We also analyze the time-complexity.

Step 1 is needed in order to consider only bimodal embeddings of G. An example of the split operation performed in Step 1 is shown in Fig. 9. Obviously, Step 1 can be executed in time $O(n+m)$, where n and m are the number of vertices and the number of edges of G, respectively. Let G' be the digraph obtained after Step 1 has been applied; we observe that all the embeddings of G' are bimodal. Also, as immediate consequence of the definition, there is a one to one correspondence between the embeddings of G' and the bimodal embeddings of G.

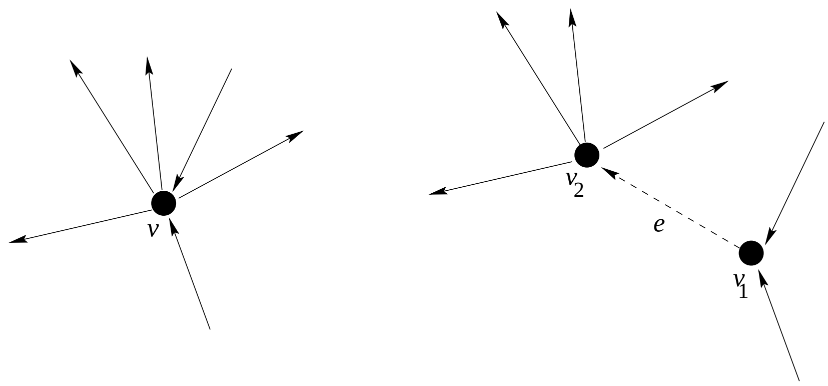


Figure 9: An example of the split vertex operation performed in Step 1 of Algorithm 2

From the above mentioned correspondence, we can assert that G' is planar if and only if G is planar bimodal. In Step 2 we execute a planarity test on G' and we find a planar embedding of G' if one exists. A planarity test (which in case determines a planar embedding) can be executed in $O(n)$ time [26, 6, 34].

If G' is not planar, we apply a "planarization" algorithm to G' (Step 3). During this step, dummy vertices are added to replace crossings between edges. Also, in order to correctly rebuild an embedding of G, we must not cross dummy edges introduced by Step 1. This can be done by applying a suitable constraint on such edges during Step 3. For example, a very simple algorithm for planarizing a graph (see [11]) is to add an edge at a time, by computing a shortest path on the dual graph to minimize the number of crossings; we can simply set an infinite cost on the dual edges corresponding to uncrossable edges, before computing the shortest path. Since a shortest path on a digraph with n vertices, and nonnegative edge-costs, can be computed in time $O(n \log n)$, if we denote by c the total number of crossings, we can execute this step in time $O(m(c+n)\log(c+n))$. The digraph G' so obtained is a planar digraph with a given planar bimodal embedding ϕ'.

In Step 5, we contract all dummy edges introduced by Step 1, restoring the original vertices that had been split. In this way, we derive a new embedded planar bimodal digraph G''_ϕ that has the same vertices as G plus c (possibly $c=0$) dummy vertices that replace crossings. Step 5 is executed in time $O(n+m)$.

In Step 6, we compute a quasi-upward planar drawing of G''_ϕ, by applying Algorithm 1. From Theorem 4.4, this step requires time $O((c+n)^2 log(c+n))$. Finally, in time $O(c)$, we remove all dummy vertices that replace crossings, in order to obtain a quasi-upward drawing of G (Step 7).

We summarize this analysis with the following theorem.

Theorem 4.5. *Let G be a digraph. There exists an* $O(m(c+n)log(c+n) + (c+n)^2 log(c+n))$

time algorithm that computes a quasi-upward drawing of G, where n and m are the number of vertices and edges of G, respectively, and c is the number of crossings of the final drawing.

Fig. 10 illustrates the steps of Algorithm 2. An example of a quasi-upward drawing computed by an implementation of Algorithm 2 is shown in Fig. 11.

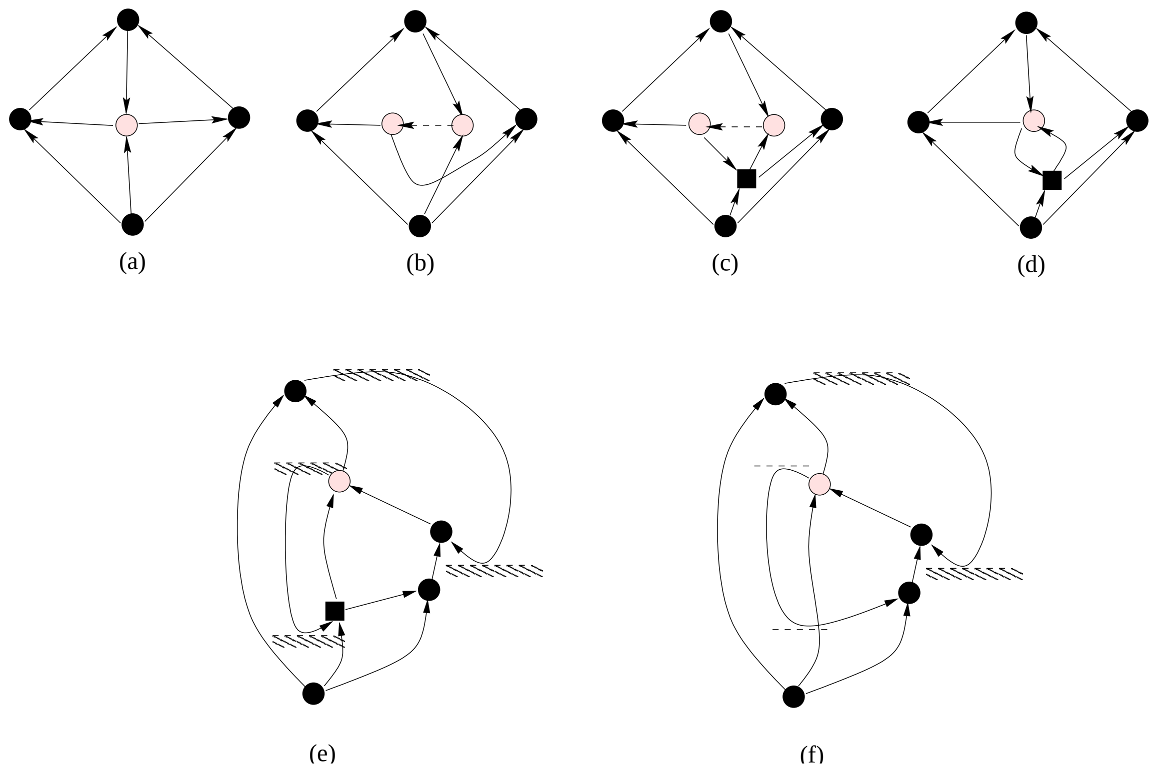


Figure 10: Illustration of the steps of Algorithm 2. (a) The input digraph. Note that the grey vertex has two incoming and two outgoing edges. (b) Splitting of vertices (Step 1). (c) A dummy vertex (represented by a black square) is introduced with a planarization technique (Steps 3 and 4). (d) The dummy edge introduced in Step 1 is contracted (Step 5). (e) A quasi-upward planar drawing is computed by Algorithm 1 (Step 6). (f) The final quasi-upward drawing is obtained by removing the dummy vertex (Step 7).

5. Changing the Planar Embeddings

Up to now we worked within a given planar embedding. In this section we consider the possibility of changing the planar embedding of the digraph, in order to find a quasi-upward planar drawing with the minimum number of bends. Observe that this problem includes the problem of testing the existence of an upward planar drawing in a variable embedding setting. Hence, it is an NP-hard problem.

We provide a branch and bound algorithm for biconnected digraphs only. When a digraph is simply connected it is possible to apply our algorithm on each biconnected component, to obtain a powerful heuristic for minimizing bends in quasi-upward drawings.

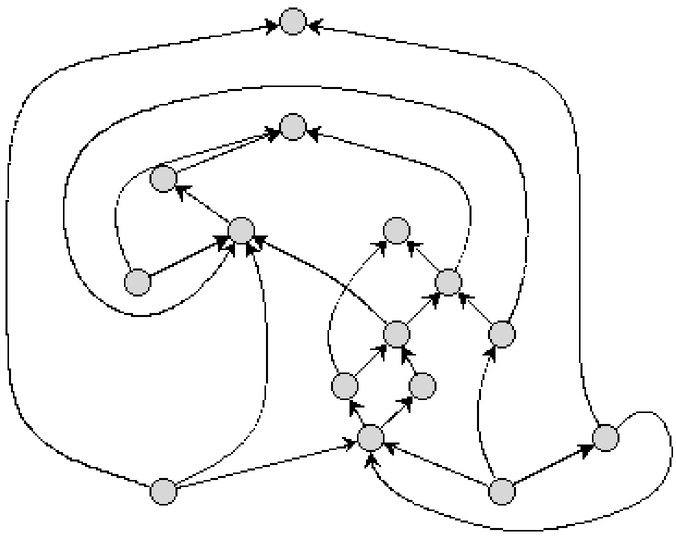


Figure 11: A quasi-upward drawing computed by an implementation of Algorithm 2.

Our branch and bound algorithm uses an enumeration schema of the planar embeddings of the digraph, based on the $SPQR$-tree data structure. Also, it takes advantage of new theorems to compute efficient lower bounds of the number of bends in quasi-upward planar drawings.

5.1. Lower Bounds on the Number of Bends of Quasi-Upward Planar Drawings

Let $G = (V, E)$ be a biconnected bimodal digraph and let Γ be a quasi-upward planar drawing of G. We denote by $b(\Gamma)$ the total number of bends of Γ, and by $b_{E'}(\Gamma)$ the number of bends on the edges of E', where $E' \subseteq E$.

Property 7. *Let $G_i = (V_i, E_i), (i = 1, \ldots, k)$ be k subgraphs of G such that $E_i \cap E_j = \emptyset$ $i \neq j$, and $\cup_{i=1,\ldots,k} E_i = E$. Let Γ_i be an optimal quasi-upward planar drawing of G_i and let Γ be an optimal quasi-upward planar drawing of G. We have:*

$$b(\Gamma) \geq \sum_{i=1,\ldots,k} b(\Gamma_i).$$

Let $G_1, \ldots, G_k$ k split components of G, and let G' be a partial digraph of G with respect to the split components $G_1, \ldots, G_k$ (See Fig. 2). Let ϕ' be an embedding of G' and let ϕ be an embedding of G that preserves ϕ'. Let $\Gamma'_{\phi'}$ be a quasi-upward planar drawing of $G'_{\phi'}$, such that $b_{E^{nonvirt}}(\Gamma'_{\phi'})$ is minimum, and let Γ_ϕ be an optimal quasi-upward planar drawing of G_ϕ. We prove the following.

Lemma 5.1.

$$b_{E^{nonvirt}}(\Gamma'_{\phi'}) \leq b_{E^{nonvirt}}(\Gamma_\phi).$$

Proof. Suppose for a contradiction that $b_{E^{nonvirt}}(\Gamma'_{\phi'}) > b_{E^{nonvirt}}(\Gamma_\phi)$. We show that this implies the existence of a quasi-upwar planar drawing of $G'_{\phi'}$ with a number of bends on the nonvirtual edge that is less than the one of $\Gamma'_{\phi'}$. For each component G_i of G ($i = 1, \dots, k$), the corresponding virtual path p_i is represented in Γ_ϕ by a simple polygonal line containing some vertices of G_i. We can construct from Γ_ϕ a quasi-upwar planar drawing $\overline{\Gamma}_{\phi'}$ of $G'_{\phi'}$ by simply deleting, for each G_i, all the edges and vertices that do not belong to p_i. Since $b_{E^{nonvirt}}(\overline{\Gamma}_{\phi'}) = b_{E^{nonvirt}}(\Gamma_\phi)$, we have that $\overline{\Gamma}_{\phi'}$ has less bends on the nonvirtual edges than $\Gamma'_{\phi'}$, a contradiction. ∎

From Property 7 and Lemma 5.1 it follows a first lower bound.

Theorem 5.2. *Let $G = (V, E)$ be a biconnected bimodal digraph and let $G'_{\phi'}$ be an embedded partial digraph of G with respect to the split components $G_1, \dots, G_k$. For each virtual path p_i of $G'_{\phi'}$, $i = 1, \dots, k$, let b_i be a lower bound on the number of bends of any quasi-upward planar drawing of the pertinent digraph G_i of p_i. Consider a quasi-upward planar drawing $\Gamma'_{\phi'}$ of $G'_{\phi'}$ such that $b_{E^{nonvirt}}(\Gamma'_{\phi'})$ is minimum. Let ϕ be an embedding of G that preserves ϕ' and consider an optimal quasi-upward planar drawing Γ_ϕ of G_ϕ. We have that:*

$$b(\Gamma_\phi) \geq b_{E^{nonvirt}}(\Gamma'_{\phi'}) + \sum_{i=1,\dots,k} b_i.$$

Property 8. *A quasi-upward planar drawing $\Gamma'_{\phi'}$ of $G'_{\phi'}$ such that $b_{E^{nonvirt}}(\Gamma'_{\phi'})$ is minimum can be computed by using Algorithm 1.*

Proof. Algorithm 1 allows to compute a quasi-upward planar drawing with the minimum number of bends of an embedded digraph, by mapping the problem into a minimum cost flow problem on a network. In our case, in order to determine $\Gamma'_{\phi'}$, when two faces f and g share a virtual path, we set the costs of the (two) edges between f and g to zero. ∎

Another lower bound is provided by the following theorem.

Theorem 5.3. *Let G_ϕ be an embedded bimodal digraph and let $G'_{\phi'}$ be an embedded partial digraph of G, such that ϕ preserves ϕ'. Consider an optimal quasi-upward planar drawing $\Gamma'_{\phi'}$ of $G'_{\phi'}$ and an optimal quasi-upward planar drawing Γ_ϕ of G_ϕ. We have that: $b(\Gamma_\phi) \geq b(\Gamma'_{\phi'})$.*

Proof. We start by observing that the partial digraph $G'_{\phi'}$ is a subgraph of G_ϕ, and that it is possible to obtain $G'_{\phi'}$ by removing a suitable set $\widetilde{E}$ of edges in G_ϕ. Suppose for a contradiction that $b(\Gamma'_{\phi'}) > b(\Gamma_\phi)$. We remove the edges in $\widetilde{E}$ from the drawing Γ_ϕ. In this way we obtain a quasi-upward planar drawing $\Gamma''_{\phi'}$ of $G'_{\phi'}$ and, of course, $b(\Gamma_\phi) > b(\Gamma''_{\phi'})$. Thus, from the assumption, we have that $b(\Gamma'_{\phi'}) > b(\Gamma''_{\phi'})$, a contradiction. ∎

The next corollary allows to combine the lower bounds of Theorem 5.2 and Theroem 5.3.

Corollary 5.4. *Let G_ϕ be an embedded biconnected bimodal digraph and $G'_{\phi'}$ an embedded partial digraph of G such that ϕ preserves ϕ'. Consider a subset F^{virt} of the set of the virtual paths of $G'_{\phi'}$. Denote by E_F the set of edges of F^{virt}. For each virtual path $p_j \notin F^{virt}$ let b_j be a lower bound on the number of bends of the pertinent digraph G_j of p_j. Consider a quasi-upward planar drawing $\Gamma'_{\phi'}$ of $G'_{\phi'}$, such that $b_{E^{nonvirt} \cup E_F}(\Gamma'_{\phi'})$ is minimum. Let Γ_ϕ be an optimal quasi-upward planar drawing of G_ϕ, we have that:*

$$b(\Gamma_\phi) \geq b_{E^{nonvirt} \cup E_F}(\Gamma'_{\phi'}) + \sum_{j:p_j \notin F^{virt}} b_j.$$

5.2. Computing Optimal Drawings with Branch and Bound Techniques

Let G be a biconnected bimodal digraph. We describe a technique for enumerating all the possible planar bimodal embeddings of G, and a strategy to avoid examining all of them in computing a quasi-upward planar drawing with the minimum number of bends. Such a technique is a variation of the one presented in [5] for orthogonal drawings.

The enumeration uses the $SPQR$-tree $\mathcal{T}$ of G. Namely, we enumerate all the planar embeddings of G with a given edge e on the external face by rooting $\mathcal{T}$ at e, and exploiting the capacity of $\mathcal{T}$ in implicitely representing such embeddings. A complete enumeration is done by considering all the possible edges and rooting $\mathcal{T}$ at all of them. Actually, in general, $\mathcal{T}$ represents also the embeddings of G that are not bimodal. To solve this problem, before computing $\mathcal{T}$, we perform on the vertices of G the split operation described in Step 1 of Algorithm 2. Dummy edges introduced by this operation will be called *straight edges*, because they cannot bend.

We encode all the possible embeddings of G with edge e on the external face with an r-uple X of variables in the following way.

- We consider the $SPQR$-tree $\mathcal{T}$ of G and root $\mathcal{T}$ at e. We visit $\mathcal{T}$ such that a node is visited after its parent, e.g. breadth first or dept first. The visit induces an ordering $\mu_1, \ldots, \mu_r$ of the P- and R-nodes of $\mathcal{T}$.

- We define an r-uple of variables $X = (x_1, \ldots, x_r)$ that are in one to one correspondence with the P- and R-nodes $\mu_1, \ldots, \mu_r$ of $\mathcal{T}$. Each x_i can have either values in the set of the nonnegative integer numbers or value "-" (null value).

  - Each variable x_i of X, corresponding to an R-node μ_i, can be set either to 0 or to 1 (corresponding to the two possible embeddings of $skeleton(\mu_i)$) or to null. The null value in variable x_i means that the embedding of $skeleton(\mu_i)$ is not specified.
  - Each variable x_j of X, corresponding to a P-node μ_j, can be set either to a value in the range $0, \ldots, (deg(\mu_j)! - 1)$ or to the null value, where $deg(\mu_j)$ is the number of children of μ_j. Again, the null value means that the embeddings of $skeleton(\mu_j)$ is not specified.

A *search tree* $\mathcal{B}$ for our problem is defined as follows.

- Each node β of $\mathcal{B}$ corresponds to a different setting X_β of $X = (x_1, \ldots, x_r)$. Such setting is partitioned into two (one of them possibly empty) subsequences $x_1, \ldots, x_h$ and $x_{h+1}, \ldots, x_r$. The elements of the first subsequence have integer values (do not have null values) while the elements in the second subsequence have null values only.

- The leaves of β are in correspondence with settings of X without null values.

- Internal nodes of β are in correspondence with settings of X with at least one null value.

- The setting of the root of β consists of null values only.

- The non-leaf node β with $X_\beta = (\overline{x}_1, \ldots, \overline{x}_h, -, \ldots, -)$ has one child for each possible integer value of x_{h+1}. The child β' of β corresponding to value $\tilde{x}_{h+1}$ has $X_{\beta'} = (\overline{x}_1, \ldots, \overline{x}_h, \tilde{x}_{h+1}, -, \ldots, -)$.

Property 9. *The leaves of $\mathcal{B}$ are in one to one correspondence with the planar embeddings of G with edge e on the external face.*

Observe that, since each internal node of $\mathcal{B}$ has at least two children, by Property 9 we have that the internal nodes of $\mathcal{B}$ are less than the number of embeddings of G with edge e on the external face.

Let G' and G'' be two different partial digraphs of G. We say that G' and G'' are *topologically equivalent* if they have the same set $E^{nonvirt}$ of edges. In other words, if G' and G'' are topologically equivalent, they differ only in the choice of the virtual paths.

Lemma 5.5. *The internal nodes of $\mathcal{B}$ are in one to one correspondence with the topologically equivalent embedded partial digraphs of G with edge e on the external face.*

Proof. Given an internal node of $\mathcal{B}$ we first show the corresponding embedded partial digraph. The embedded partial digraph G_β of G associated with node β of $\mathcal{B}$ with $X_\beta = (\overline{x}_1, \ldots, \overline{x}_h, -, \ldots, -)$ is obtained as follows. First, set G_β to $skeleton(\mu_1)$ embedded according to $\overline{x}_1$. Second, substitute each virtual edge $e_i (2 \leq i \leq h)$ of μ_1 with the skeleton of the child μ_i of μ_1, embedded according to $\overline{x}_i$. Then, recursively substitute virtual edges with embedded skeletons until all the skeletons in $\{skeleton(\mu_1), \ldots, skeleton(\mu_h)\}$ have been used.

The fact that the embedded partial digraphs associated with two distinct nodes β_1 and β_2 are distinct follows immediately from the presence of at least one different value in X_{β_1} and X_{β_2}.

Finally, let G' be an embedded partial digraph of G in a class of topologically equivalent partial digraphs, with edge e on the external face. We can assume that G' is the partial digraph of the class such that all its virtual paths are virtual paths in some skeletons of the nodes of the $SPQR$-tree. The fact that there is a node of $\mathcal{B}$ associated with G' follows immediately from the definition of $\mathcal{B}$. ∎

Our computation works as follows. We visit $\mathcal{B}$ breadth-first starting from the root. At each internal node β of $\mathcal{B}$ with setting X_β we compute a lower bound and an upper bound of the number of bends of any quasi-upward drawing of G with the embedding (partially) specified by X_β. The current optimal solution is updated accordingly. The children of β are not visited if the lower bound is greater than the current optimum.

For each β, lower bounds and upper bounds are computed as follows, by using the results presented in Section 5.1.

1. We construct the embedded partial digraph G_β corresponding to β (see Lemma 5.5).

2. We compute lower bounds.

   - Let $E^{nonvirt}$ be the set of nonvirtual edges of G_β. For each virtual path p_i of G_β consider the pertinent digraph G_i and a lower bound b_i on the number of bends of any quasi-upward planar drawing of G_i. See Theorem 5.2. Denote by F^{virt} the set of virtual paths p_i such that $b_i = 0$, and by E_F the set of edges of such paths.
   - By Property 8 we can apply Algorithm 1 on G_β, assigning zero costs to the arcs of the flow network between faces that share a virtual path of G_β that is not in F^{virt}. Also, in order to have no bend on the straight edges we set the cost of the corresponding arcs in $\mathcal{N}$ to infinity. Observe that, after this setting the flow problem remains feasible. In fact, the straight edges cannot cause cycles in any partial digraph.

We obtain a quasi-upward planar drawing Γ_β of G_β with the minimum number of bends on the set $E^{nonvirt} \cup E_F$. Let $b_{E^{nonvirt} \cup E_F}(\Gamma_\beta)$ be such a number of bends. Then, by Corollary 5.4 we compute the lower bound L_β at node β, as

$$L_\beta = b_{E^{nonvirt} \cup E_F}(\Gamma_\beta) + \sum_{i: p_i \notin F^{virt}} b_i$$

Lower bounds b_i can be pre-computed with a suitable pre-processing strategy that will be illustrated afterwards.

3. We compute upper bounds. Namely, we consider the embedded partial digraph G_β and complete it to a pertinent embedded digraph G_ϕ. The embedding of G_ϕ is obtained by substituting the null values of X_β with integer values choosen in a random way over all the possible values. Then we apply Algorithm 1 to G_ϕ so obtaining an upper bound. We also avoid multiple generations of the same embedded digraph in completing the partial digraph.

The pre-computation of the b_i lower bounds is done in two phases:

1. For each P- and R-node μ_i we apply several times Algorithm 1, to compute a quasi-upward drawing with the minimum number of bends of $skeleton(\mu_i)$. The algorithm is applied for all the possible embeddings of $skeleton(\mu_i)$ with the reference edge on the external face. Also, the bends on the virtual edges are considered as zero cost bends. We choose the drawing with the minimum number of bends to get a correct lower bound. Let $\bar{b}_i$ such a number. We label μ_i with $\bar{b}_i$. For each Q- and S-node μ_i, $\bar{b}_i = 0$.
2. We visit $\mathcal{T}$ bottom-up. At each node μ_i we compute b_i as the sum of $\bar{b}_i$ plus the values of the b_j for each child μ_j of μ_i.

Of course, to compute the optimal solution, we must apply the above algorithm over all the possible choices of the reference edge e. For each choice we root $\mathcal{T}$ at a different Q-node. Observe that this strategy can lead to consider several times the same embedding. To avoid this, if an edge e that has already been reference edge appears on the external face of some G_β under computation, since we have already explored all the embeddings with e on the external face, we cut from the search tree the subtree rooted at β. In this way each embedding is considered exactly once.

5.3. Speed-up of the Branch and Bound algorithm

In this section we illustrate techniques that can be used to improve the quality of the lower bounds and to discard specific subsets of planar embeddings through the computation.

The first observation we can do is that "choosing virtual paths with a few switches is better". In fact, a switch along a path p augments the degrees of freedom of p in "turning" without generating bends. More formally, we can show the following.

Lemma 5.6. *Let G be a planar bimodal digraph and let $G'_{\phi'}$ and $G''_{\phi''}$ two topologically equivalent partial digraphs of G, that differ only in the choice of a virtual path in a split component G_i of G, and such that ϕ preserves ϕ' and ϕ''. Denote by p'_i and p''_i the paths replacing G_i in $G'_{\phi'}$ and $G''_{\phi''}$, respectively, and suppose p'_i has less switches than p''_i. Let $\Gamma'_{\phi'}$ and $\Gamma''_{\phi''}$ be two quasi-upward planar drawings, with the minimum number of bends, of $G'_{\phi'}$ and $G''_{\phi''}$, respectively. Then $b(\Gamma'_{\phi'}) \geq b(\Gamma''_{\phi''})$.*

Proof. We observe that, in a polygonal line of a quasi-upward drawing, we can replace a bend point with a source or sink switch, without changing the shape of the polygonal line, and so reducing its number of bends. From this fact, we can derive from $\Gamma'_{\phi'}$ a quasi-upward planar drawing of $G''_{\phi''}$ with a number of bends that is less or equal than the number of bends of $\Gamma'_{\phi'}$, by substituting path p'_i with any path having more switches, for example p''_i. In this way we obtain a quasi-upward planar drawing of $G''_{\phi''}$ with a number of bends greater or equal than the $\Gamma''_{\phi''}$ one, because $\Gamma''_{\phi''}$ is an optimal solution. Hence the assertion. ∎

Because of Lemma 5.6, we are interested to choose virtual paths with the minimum number of switches. This can be done in linear time by using the following result.

Theorem 5.7. *Let G be a digraph with n vertices and m edges and let u and v be two distinct vertices of G. A simple path with the minimum number of switches between u and v can be found in $O(n+m)$ time.*

Proof. We first describe an algorithm that computes a path with a minimum number of switches between u and v, then we show the correctness of the algorithm and its linear time complexity.

We call *forward (backward)* a depth-first-search on a digraph in which we can move from a vertex along outgoing (incoming) edges only. Also, we call *opposite* the two different kinds of depth-first-search. The algorithm to find a path with minimum number of switches between u and v is as follows:

1. We first execute a visit of the digraph starting from u, consisting of an alternate sequence of forward and backward depth-first-searches. Each search is called *step.* Steps are numbered starting from 0. Step 0 performs a forward depth-first-search. Then, we execute a second visit of the digraph, again starting from u with the same alternating technique. In the second visit Step 0 performs a backward depth-first-search.

2. In step 0 of any of the two visits, we find all vertices reachable from u through a path having 0 switches, starting with one of the two possible directions (forward or backward). We denote by V_0 the set of such vertices. At this point, we change the direction used in the previous depth-first-search, and we execute a series of depth-first-searches (opposite with respect the first one) each one starting from a different vertex of V_0. In this way we find all vertices reachable from u through a path having 1 switch, starting with one of the two possible directions.

3. Suppose that in step $k-1$ we have found all the vertices reachable from u through a path having $k-1$ switches, starting with one of the two possible directions. We denote by V_{k-1} the set of such vertices. In step k we execute a series of depth-first-searches, opposite with respect the previous ones, each one starting from a different vertex of V_{k-1}.

4. We iterate the above procedure k times, until v belongs to V_k.

5. Suppose that v belongs to the set V_{k_1} of the first visit (starting with a forward depth-first-search), and suppose that v belongs to the set V_{k_2} of the second visit (starting with a backward depth-first-search). The length of the path with minimum number of switches is the minimum between k_1 and k_2.

We prove the correctness of the algorithm by induction. It is sufficient to show the correctness of one of the two visits, for example the first one (which starts with a forward depth-first-search). It is obvious that in step 0 we reach all and only the vertices w such that there is a directed (forward) path from u to w, that is a path with 0 switches. Suppose that, in step $k-1$, we reach all and only the vertices w such that there is a path from u to w having $k-1$ switches. We prove that this holds also for step k, by applying the above procedure. Suppose w_k is a vertex in V_k. Of course, there is a 0 switches path from a vertex of V_{k-1} and w_k in one of the two possible directions. Then, there is a path with k switches from u to w_k. Also, there is not a path from u to w_k that has a number of switches less than k, because otherwise it would have been found in a previous step. Finally, we have to show that does not exist a vertex w such that w is reachable from u through a k switches path, and w does not belong to V_k. If such a vertex exists, then it would be reachable from a vertex of V_{k-1} by a 0 switches path, in one of the two possible directions, and then it would belong to V_k, a contradiction.

In order to prove that the algorithm has an $O(n+m)$ time complexity, it is sufficient to observe that each edge is visited exactly once for each of the two distinct visits. ∎

The second observation we can do allows to reduce the set of planar embeddings that we need to explore during the branch and bound computation. Namely, suppose $e_1, e_2, \ldots, e_k$ is a set of multiple edges with the same direction between two vertices u and v. We can consider only the planar embeddings in which such edges appear consecutively in a circular ordering around both vertices u and v. This corresponds to collapsing these edges in one only edge, while considering all permutations of the children of P-nodes in the $SPQR$-tree. The correctness of this observation is proved in the following theorem.

Theorem 5.8. *Let G be a biconnected planar bimodal digraph and let e_1 and e_2 be two multiple edges with the same direction in G. For each quasi-upward planar drawing Γ' of G in which e_1 and e_2 are not consecutive around their end-vertices, there exists a quasi-upward planar drawing Γ of G such that e_1 and e_2 are consecutive around their end-vertices, and $b(\Gamma) \leq b(\Gamma')$.*

Proof. Let b_1 and b_2 be the number of bends on edges e_1 and e_2, respectively, in Γ'. Without loss of generality, we can assume that $b_1 \leq b_2$. We can derive from Γ' a quasi-upward planar drawing Γ with a number of bends less or equal than the number of bends of Γ'. In fact, in Γ' we first remove edge e_2, and then we draw a polyline curve (with the same shape as edge e_1) close to e_1 and labeled e_2. This is always possible by simple geometric considerations (see Fig. 12). In this way we obtain a quasi-upward planar drawing Γ such that e_1 and e_2 are consecutive around their extremal vertices and such that $b(\Gamma) \leq b(\Gamma')$. ∎

6. Experimental Results

All the algorithms described in this paper have been implemented and extensively tested. In particular we tested the branch and bound algorithm described in Section 5 over a randomly generated test suite, consisting of about 300 digraphs. The test suite is available on the Web (`www.dia.uniroma3.it/~gdt`). For generating the digraphs we started from the following result. Any embedded planar biconnected graph can be generated from a triangle graph, by means of a sequence of *insert-vertex* and *insert-edge* operations [15]. Insert-vertex operation subdivides an existing edge into two edges separated by a new vertex. Insert-edge operation inserts a new edge between two existing vertices that are on the border of a same face. We implemented a generation mechanism that at each step randomly chooses which operation to apply and where

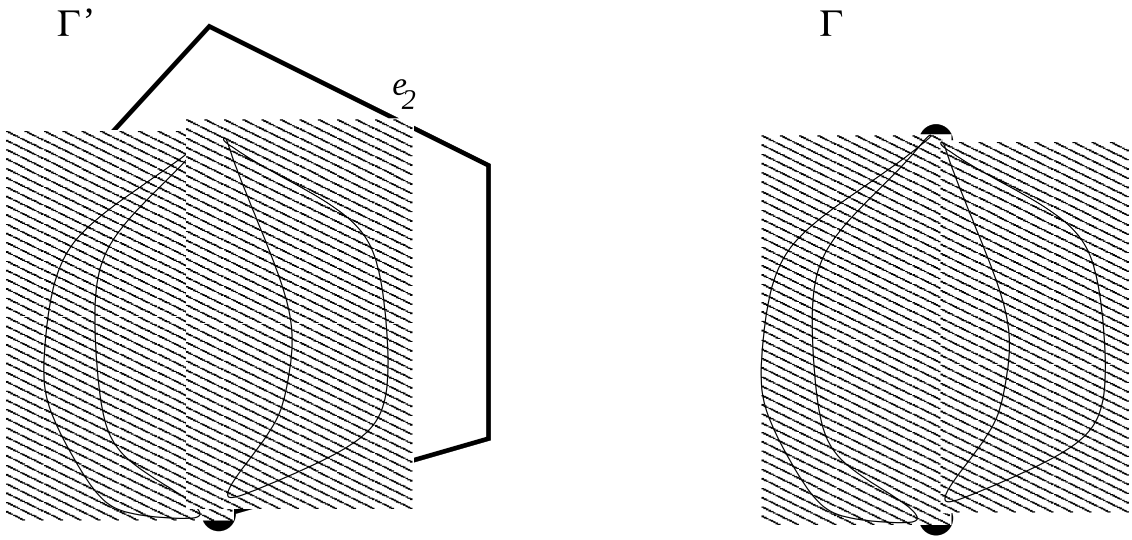


Figure 12: Illustration of the proof of Theorem 5.8

to apply it. After the generation, edges are randomly oriented, and then the digraph is discarded if it is not bimodal. The density (number of edges over number of vertices) of the generated digraphs is in the range 1.2–2, and the number of vertices is in the range 10–200.

The implementation uses on the C++ language and the GDToolkit graph drawing library (`www.dia.uniroma3.it/~gdt`). Experiments have been done with a Sun Ultrasparc 1. The results of the experiments are summarized in the graphics of Fig. 13. To check the applicability of the algorithm we measured the CPU time (Fig. 13.a). To better understand the features of the test suite we measured the number of embeddings (Fig. 13.b) and the number of components affecting the computation time (Fig. 13.c).

We observe that the computation takes an average time that is less than 20 minutes for digraphs with a high number of vertices. We also observe that we tested digraphs that have up to $1,000,000$ of planar embeddings.

7. Conclusions and Open Problems

We presented a new approach in constructing drawings of digraphs. Such approach can be considered as an equivalent of the popular topology-shape-metrics approach (that constructs orthogonal drawings of undirected graphs; see e.g. [39, 40]) for drawing digraphs. In fact, the drawing process presented in this paper can be seen as a sequence of steps. During the first step a topology is found in terms of a bimodal planar embedding of the digraph. Dummy vertices representing crossings may be inserted. During the second step a shape is found with the minimum number of bends (within the given topology) in terms of an intermediate representation of the drawing. During the last step the final drawing is constructed by using any of the technique for drawing planar *st*-digraphs (see [10]) and, eventually, the bends are smoothed. Further, for the applications for which to have a tidy drawing it is really important, even at the expense of a higher computation time, we have shown a branch and bound technique that minimizes

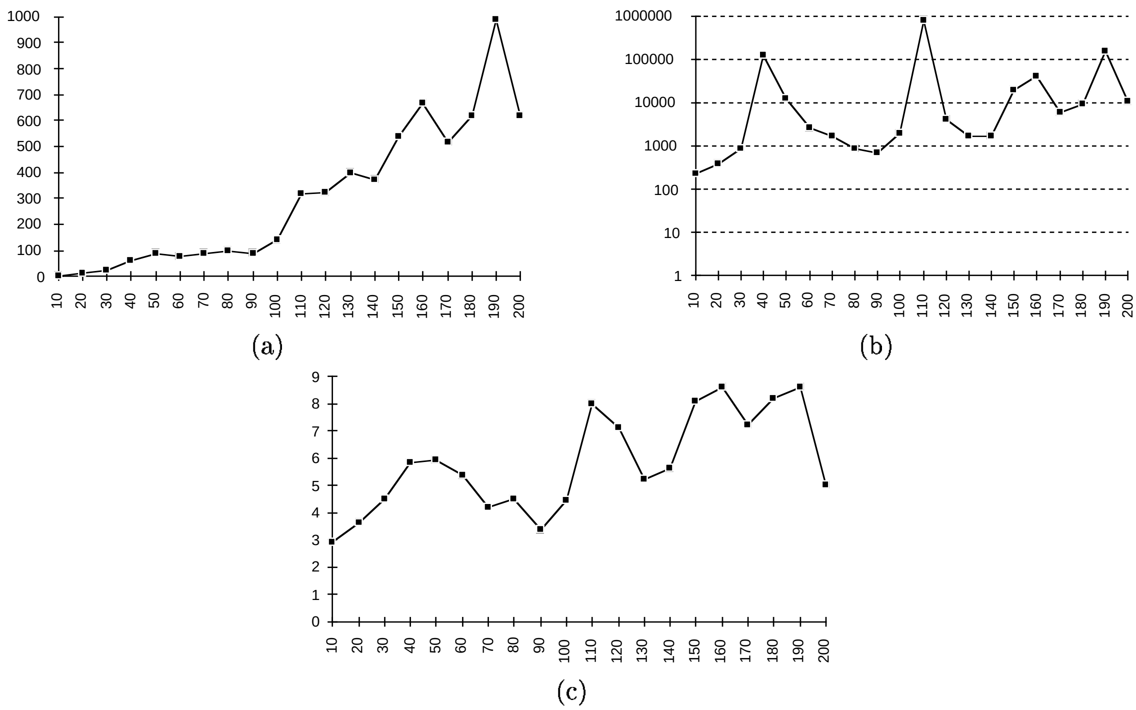


Figure 13: Graphics summarizing the experiments: (a) CPU time (seconds), (b) number of embeddings (log. scale), and (c) number of components (sum of P and R nodes). The x-axis represents the number of vertices and in the y-axis we give average values.

the number of bends of each biconnetced component, by searching all the possible topologies of the component. We have also shown that the algorithm has a reasonable time performance for digraphs with up to 200 vertices.

This paper also opens several problems related to quasi-upward planarity.

Is there a relationship between the minimum-feedback arc set problem [19] and the minimization of the number of bends? How is such minimization related to the search of a maximum upward planar subgraph of a given digraph?

Acknowledgements

We are grateful to Antonio Leonforte for his help in the implementation. We are also grateful to Sandra Follaro, Maurizio Patrignani, and Maurizio Pizzonia for their support.

References

[1] `http://www.daimi.au.dk/PetriNets/people/form.html`.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network flows," in *Optimization* (G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds.), vol. 1 of *Handbooks in Operations Research and Management*, pp. 211–360, North-Holland, 1990.

[3] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino, "Upward drawings of triconnected digraphs," *Algorithmica*, vol. 6, no. 12, pp. 476–497, 1994.

[4] P. Bertolazzi, G. Di Battista, C. Mannino, and R. Tamassia, "Optimal upward planarity testing of single-source digraphs," in *Proc. 1st Annu. European Sympos. Algorithms*, vol. 726 of *Lecture Notes Comput. Sci.*, pp. 37–48, Springer-Verlag, 1993.

[5] P. Bertolazzi, G. Di Battista, and W. Didimo, "Computing orthogonal drawings with the minimum number of bends," in *Proc. 5th Workshop Algorithms Data Struct.* (F. Dehne, A. Rau-Chaplin, J.-R. Sack, and R. Tamassia, eds.), vol. 1272 of *Lecture Notes Comput. Sci.*, pp. 331–344, Springer-Verlag, 1997.

[6] K. Booth and G. Lueker, "Testing for the consecutive ones property interval graphs and graph planarity using PQ-tree algorithms," *J. Comput. Syst. Sci.*, vol. 13, pp. 335–379, 1976.

[7] N. Cancedda, G. W. Karnstrup, E. Pianta, and E. Pietrosanti, "Sax: Generating hypertext from sadt models," in *Workshop on Applications of Natural Language to Information Systems* (G. Italiano and S. Orlando, eds.), 1997. `http://ecate.itc.it:1024/kamstrup/sax/article2/index-nldb.html`.

[8] P. Crescenzi, G. Di Battista, and A. Piperno, "A note on optimal area algorithms for upward drawings of binary trees," *Comput. Geom. Theory Appl.*, vol. 2, pp. 187–200, 1992.

[9] P. Crescenzi and A. Piperno, "Optimal-area upward drawings of AVL trees," in *Graph Drawing (Proc. GD '94)* (R. Tamassia and I. G. Tollis, eds.), vol. 894 of *Lecture Notes Comput. Sci.*, pp. 307–317, Springer-Verlag, 1995.

[10] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, "Algorithms for drawing graphs: an annotated bibliography," *Comput. Geom. Theory Appl.*, vol. 4, pp. 235–282, 1994.

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

[12] G. Di Battista, W. P. Liu, and I. Rival, "Bipartite graphs upward drawings and planarity," *Inform. Process. Lett.*, vol. 36, pp. 317–322, 1990.

[13] G. Di Battista and R. Tamassia, "Algorithms for plane representations of acyclic digraphs," *Theoret. Comput. Sci.*, vol. 61, pp. 175–198, 1988.

[14] G. Di Battista and R. Tamassia, "On-line maintenance of triconnected components with SPQR-trees," *Algorithmica*, vol. 15, pp. 302–318, 1996.

[15] G. Di Battista and R. Tamassia, "On-line planarity testing," *SIAM J. Comput.*, vol. 25, pp. 956–997, 1996.

[16] G. Di Battista, R. Tamassia, and I. G. Tollis, "Area requirement and symmetry display of planar upward drawings," *Discrete Comput. Geom.*, vol. 7, no. 4, pp. 381–401, 1992.

[17] S. Even, *Graph Algorithms.* Potomac, Maryland: Computer Science Press, 1979.

[18] U. Fößmeier and M. Kaufmann, "On bend-minimum orthogonal upward drawing of directed planar graphs," in *Graph Drawing (Proc. GD '94)* (R. Tamassia and I. G. Tollis, eds.), vol. 894 of *Lecture Notes Comput. Sci.*, pp. 52–63, Springer-Verlag, 1995.

[19] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York, NY: W. H. Freeman, 1979.

[20] A. Garg, M. T. Goodrich, and R. Tamassia, "Planar upward tree drawings with optimal area," *Internat. J. Comput. Geom. Appl.*, vol. 6, pp. 333–356, 1996.

[21] A. Garg and R. Tamassia, "Efficient computation of planar straight-line upward drawings," in *Graph Drawing '93 (Proc. ALCOM Workshop on Graph Drawing)*, 1993.

[22] A. Garg and R. Tamassia, "On the computational complexity of upward and rectilinear planarity testing," in *Graph Drawing (Proc. GD '94)* (R. Tamassia and I. G. Tollis, eds.), vol. 894 of *Lecture Notes Comput. Sci.*, pp. 286–297, Springer-Verlag, 1995.

[23] A. Garg and R. Tamassia, "Upward planarity testing," *Order*, vol. 12, pp. 109–133, 1995.

[24] S. M. Hashemi and I. Rival, "Upward drawings to fit surfaces," in *Proc. Workshop on Orders, Algorithms and Applications*, vol. 831 of *Lecture Notes Comput. Sci.*, pp. 53–58, Springer-Verlag, 1994.

[25] J. Hopcroft and R. E. Tarjan, "Dividing a graph into triconnected components," *SIAM J. Comput.*, vol. 2, no. 3, pp. 135–158, 1973.

[26] J. Hopcroft and R. E. Tarjan, "Efficient planarity testing," *J. ACM*, vol. 21, no. 4, pp. 549–568, 1974.

[27] M. D. Hutton and A. Lubiw, "Upward planar drawing of single-source acyclic digraphs," *SIAM J. Comput.*, vol. 25, no. 2, pp. 291–311, 1996.

[28] M. Jünger, S. Leipert, and P. Mutzel, "Pitfalls of using PQ-Trees in automatic graph drawing," in *Graph Drawing (Proc. GD '97)* (G. Di Battista, ed.), vol. 1353 of *Lecture Notes Comput. Sci.*, pp. 193–204, Springer-Verlag, 1997.

[29] D. Kelly, "Fundamentals of planar ordered sets," *Discrete Math.*, vol. 63, pp. 197–216, 1987.

[30] D. Kelly and I. Rival, "Planar lattices," *Canad. J. Math.*, vol. 27, no. 3, pp. 636–665, 1975.

[31] S. K. Kim, "Simple algorithms for orthogonal upward drawings of binary and ternary trees," in *Proc. 7th Canad. Conf. Comput. Geom.*, pp. 115–120, 1995.

[32] X. Lin and P. Eades, "Area requirements for drawing hierarchically planar graphs," in *Graph Drawing (Proc. GD '97)* (G. Di Battista, ed.), vol. 1353 of *Lecture Notes Comput. Sci.*, pp. 219–229, Springer-Verlag, 1997.

[33] B. Mohar and P. Rosenstiehl, "A flow approach to upward drawings of toroidal maps," in *Graph Drawing (Proc. GD '94)* (R. Tamassia and I. G. Tollis, eds.), vol. 894 of *Lecture Notes Comput. Sci.*, pp. 33–39, Springer-Verlag, 1995.

[34] T. Nishizeki and N. Chiba, "Planar graphs: Theory and algorithms," *Ann. Discrete Math.*, vol. 32, 1988.

[35] A. Papakostas, "Upward planarity testing of outerplanar dags," in *Graph Drawing (Proc. GD '94)* (R. Tamassia and I. G. Tollis, eds.), vol. 894 of *Lecture Notes Comput. Sci.*, pp. 298–306, Springer-Verlag, 1995.

[36] C. Platt, "Planar lattices and planar graphs," *J. Combin. Theory Ser. B*, vol. 21, pp. 30–39, 1976.

[37] E. Reingold and J. Tilford, "Tidier drawing of trees," *IEEE Trans. Softw. Eng.*, vol. SE-7, no. 2, pp. 223–228, 1981.

[38] C.-S. Shin, S. K. Kim, and K.-Y. Chwa, "Area-efficient algorithms for upward straight-line tree drawings," in *Proc. 2nd International Computing and Combinatorics: COCOON'96*, vol. 1090 of *Lecture Notes Comput. Sci.*, pp. 106–116, Springer-Verlag, 1996.

[39] R. Tamassia, "On embedding a graph in the grid with the minimum number of bends," *SIAM J. Comput.*, vol. 16, no. 3, pp. 421–444, 1987.

[40] R. Tamassia, G. Di Battista, and C. Batini, "Automatic graph drawing and readability of diagrams," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-18, no. 1, pp. 61–79, 1988.

[41] L. Trevisan, "A note on minimum-area upward drawing of complete and Fibonacci trees," *Inform. Process. Lett.*, vol. 57, no. 5, pp. 231–236, 1996.