Quasi-Upward Planarity*
(Extended Abstract)

Paola Bertolazzi[1], Giuseppe Di Battista[2], and Walter Didimo[2]

[1] IASI, CNR, viale Manzoni 30, 00185 Roma Italy. bertola@iasi.rm.cnr.it
[2] Dipartimento di Informatica e Automazione, Università di Roma Tre
via della Vasca Navale 79, 00146 Roma, Italy. {gdb,didimo}@dia.uniroma3.it

Abstract. In this paper we introduce the quasi-upward planar drawing convention and give a polynomial time algorithm for computing a quasi-upward planar drawing with the minimum number of bends within a given planar embedding. Further, we study the problem of computing quasi-upward planar drawings with the minimum number of bends of digraphs considering all the 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 a planar upward drawing.

Planar upward 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 [10]). Upward planarity of specific families of digraphs has been studied in: planar *st*-digraphs [14, 7], embedded and triconnected digraphs [3], single source digraphs [13, 4], bipartite digraphs [6], outerplanar digraphs [17], trees [18], and hierarchical digraphs [15]. The NP-completeness of upward planarity testing is proved in [9]. Further, an impressive set of results on upward drawings can be found in the literature on ordered sets.

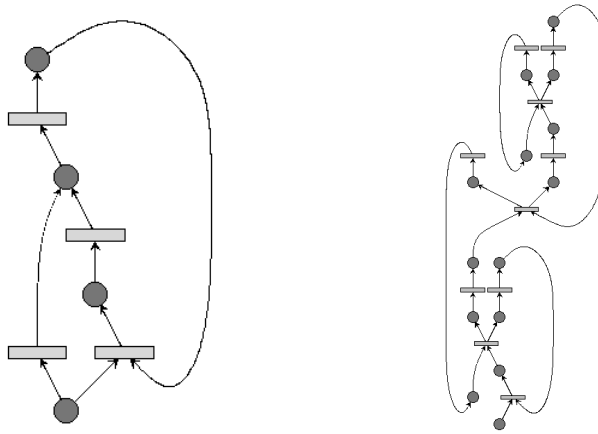
Despite such a long list of results, upward planar drawings have found limited applicability. The reasons for this are mainly in the tightness of the upward planar standard that can be satisfied for "a few" digraphs. Also, the applications require very often a similar but slightly different standard, where the drawing is upward "as much as possible". This is the case, for example, of Petri Nets or of certain types of SADT diagrams.

In this paper we introduce and investigate quasi-upward planar drawings. A *quasi-upward drawing* Γ of a digraph is such that the horizontal line through each vertex "locally splits" the incoming edges from the outgoing edges. More formally, for each vertex v there exists a connected region R of the plane, properly

* Research supported in part by the ESPRIT LTR Project no. 20244 - ALCOM-IT and by the CNR Project "Geometria Computazionale Robusta con Applicazioni alla Grafica ed al CAD."

S.H. Whitesides (Ed.): GD'98, LNCS 1547, pp. 15–29, 1998.

containing v, such that, in the intersection of R with Γ, the horizontal line through v separates the incoming edges from the outgoing edges. Examples of quasi-upward (planar) drawings of Petri Nets are shown in Fig. 1.



(a) (b)

Fig. 1. Two quasi-upward drawings of Petri Nets.

Observe that an upward drawing is quasi-upward 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 give a polynomial time algorithm for computing a quasi-upward planar drawing with the minimum number of bends of an embedded "bimodal" digraph and show how to extend the technique to deal with non-bimodal digraphs; we use a min-cost flow technique that unifies the techniques for orthogonal drawings presented in [19] with those presented in [3] for upward planarity (Section 3).

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 digraphs considering all the possible planar embeddings. Thus, we present: lower bounds techniques for quasi-upward planar drawings (Section 4); a branch-and-bound algorithm for computing a quasi-upward planar drawing with the minimum number of bends of a biconnected digraph

(Section 5). Such a technique is a variation of the technique presented in [2] for orthogonal drawings and can be used for each biconnected component of a digraph, constituting the basis of a powerful drawing heuristic. Further, it 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 Preliminaries

We assume familiarity with planarity and connectivity of graphs [16]. Since we consider only planar graphs, we use the term *embedding* instead of *planar embedding*. The following definitions, usually introduced for graphs, are used here for digraphs. Let G be a biconnected digraph. A *split pair* of G is either a separation-pair or a pair of adjacent vertices. A *split component* of a split pair $\{u, v\}$ 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 split component of $\{u, v\}$. Suppose $G_1, \ldots, G_k$ are some pairwise edge disjoint split components of G with split pairs $u_1, v_1 \ldots u_k, v_k$, respectively. The digraph $G' \subseteq G$ obtained by substituting each G_i $(i = 1, \ldots, k)$ with any simple path p_i between u_i and v_i in G_i is a *partial digraph* of G. Paths p_i are called *virtual paths*. We denote E^{virt} the set of the edges of the virtual paths of G' and we denote $E^{nonvirt}$ the set of the edges of G' that are not in the virtual paths. We say that G_i is the *pertinent digraph* of p_i and that p_i is the *representative path* of G_i.

Let ϕ be an embedding of G and ϕ' 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.

In the following we revise SPQR-trees [8], 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 [12]. 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.

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 μ 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 $skeleton(\mu)$ is replaced by any simple path (*virtual path*) of the pertinent digraph of (u, v) between u and v. Also, we call *reference path* the path that substitutes the reference edge. Observe that, after this modification, $skeleton(\mu)$ is directed and is a subgraph of G.

An SPQR-tree $\mathcal{T}$ rooted at a given Q-node represents all the planar embeddings of G having the reference edge (associated to the Q-node at the root) on the external face.

3 Quasi-Upward Planarity

Let G be an embedded planar digraph. A vertex of G is *bimodal* if its incident list can be partitioned into two possibly empty linear lists one consisting of incoming edges and the other consisting of outgoing edges. If all its vertices are bimodal then G and its embedding are called *bimodal*. A digraph is *bimodal* if it has a planar bimodal embedding.

We define the operation *insert-switch* on an edge of an embedded digraph. Operation $2k$-insert-switch on edge (u, v) removes (u, v) and inserts vertices $v_1, u_1, v_2, u_2, \ldots, v_k, u_k$ and edges (u, v_1), (u_1, v_1), (u_1, v_2), $(u_2, v_2), \ldots, (u_k, v)$. See Fig. 2. The new path is embedded in place of (u, v). Observe that vertices $u_1, \ldots, u_k$ $(v_1, \ldots, v_k)$ are k new sources (sinks) of the digraph.

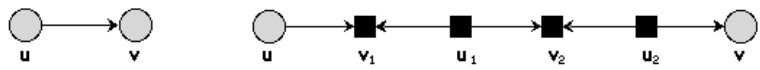


Fig. 2. An example of 4-insert-switch.

Let G be a bimodal digraph that is not upward planar and let G' be a digraph obtained from G by performing one $2k$-insert-switch operation on (u, v). Suppose that G' is upward planar (and, of course, acyclic) and consider an upward drawing Γ' of G'. Reverse the direction of edges $(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)$ of Γ'. We have that Γ' is a quasi-upward drawing with $2k$ bends of G, where edge (u, v) of G is "represented" by the drawing of path $v_1, u_1, v_2, u_2, \ldots, v_k, u_k$ of G'. Each of $v_1, u_1, v_2, u_2, \ldots, v_k, u_k$ corresponds to a bend.

In Fig. 3.a we show a bimodal digraph that is not upward planar. Observe that if we perform a 2-insert-switch operation on edge $(3, 2)$, then the resulting digraph becomes upward planar. See Fig. 3.b. Observe that the resulting drawing can be simply modified into a quasi-upward planar drawing of the original digraph (Fig. 3.c) by reversing one edge. The best aesthetic results are obtained by smoothing the bends (Fig. 3.d).

We consider the following problem. Given a bimodal digraph G we want to determine a quasi-upward planar drawing of G with the minimum number of bends. In this section we restrict our attention to a given planar embedding, while in the next sections we shall consider also the possibility of changing the embedding.

Let G be an embedded bimodal planar digraph and let S (T) be the set of its sources (sinks). Let f be a face of G. We visit the contour of f counterclockwise (i.e. such that the face remains always to the left during the visit). Let $2n_f$ be the number of pairs of consecutive edges e_1, e_2 such that the the direction of e_1 is opposite to the direction of e_2. The *capacity* c_f of f is $n_f - 1$ if f is an internal face and $n_f + 1$ if f is the external face.

Lemma 1 and Theorem 1 have been shown in [3].

Lemma 1. *The sum of the capacities of all the faces is equal to* $|S| + |T|$.

An assignment of the sources and sinks of G to the faces such that the following properties hold is *upward consistent*: (1) a source (sink) is assigned to

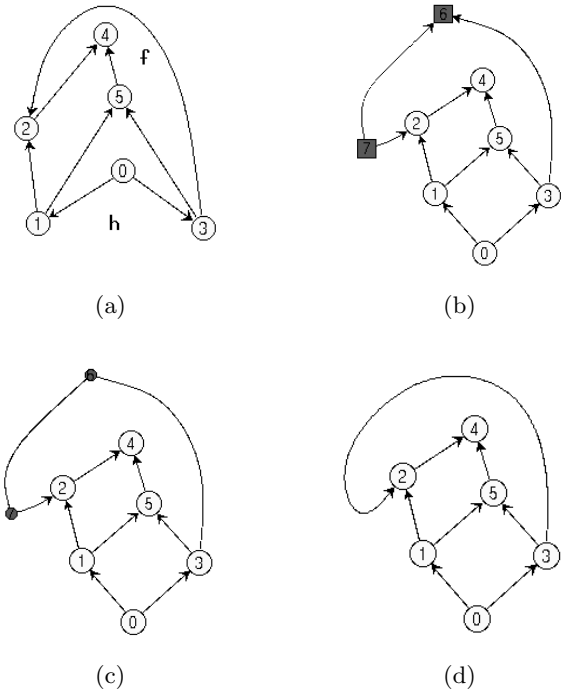


Fig. 3. (a) A non upward planar digraph; (b) An upward planar drawing after a 2-insert-switch operation; (c) A quasi-upward planar drawing; and (d) A quasi-upward planar drawing with smoothed bends.

exactly one of its incident faces; for each face f, the number of sources and sinks that are assigned to f is equal to c_f.

Theorem 1. *Let G be an embedded bimodal digraph; G is upward planar if and only if it admits an upward-consistent assignment.*

Consider again Fig. 3. If we interpret the 2-insert-switch in terms of capacity and assignment we have that before the insertion (Fig. 3.a) the capacity of the external face h was equal to 2 while the capacity of all the internal faces was equal to 0. Also, the external face had one source (vertex 0) to accomodate while face f had one sink (vertex 4). Hence, we had a surplus of capacity on h and a deficiency of capacity on f. The effect of the insertion was to increase the capacity of both f and h of one unit. At the same time we have now two more sources and sinks to assign. However, such vertices can be both assigned to h.

A maximal path whose edges share two (not necessarily distinct) faces is a *boundary* path.

We extend the concept of upward consistent assignment, that has been used for upward planarity, by introducing a new flow network that models quasi-

upward planarity of a digraph within a given planar embedding. Namely, each quasi-upward planar drawing corresponds to a flow in the network. Further, each flow corresponds to an equivalence class of quasi-upward planar drawings. For each element of the equivalence class we have the same number of bends along each boundary path. Also, for each sink t, consider the horizontal line λ through t and a sufficiently small region R properly enclosing t; the intersection between R and the halfplane "above" λ is a subset of the same face for each drawing of the class; and for each source s, consider the horizontal line λ through s and a sufficiently small region R properly enclosing s; the intersection between R and the halfplane "below" λ is a subset of the same face for each drawing of the class.

The flow network $\mathcal{N}$ associated to an embedded bimodal digraph G is defined as follows: nodes have supplies and demands; arcs have a capacity β and a cost χ; the nodes of $\mathcal{N}$ are the sources, the sinks, and the faces of G; a source or sink node v produces a flow $\sigma(v) = 1$; a face-node f consumes a flow $\sigma(f) = c_f$, where c_f is the capacity of f. Observe that if f is internal and is a directed cycle then $c_f = -1$. This corresponds to a production of flow rather than a consumption; for each vertex-node v we have one arc from v to every its incident face f. Such arcs have zero cost and capacity $\beta(v, f) = 1$; for each boundary path between faces f and g we have arcs (f, g) and (g, f). Arcs have cost $\chi(f, g) = \chi(g, f) = 2$ and a capacity $\beta(f, g) = \beta(g, f) = +\infty$.

By Lemma 1 it follows that the sum of the amounts of flow supplied by the vertex-nodes is equal to the sum of the amounts of flow consumed by the face-nodes. Considering that arcs between face-nodes have infinite capacity we have:

Lemma 2. *Network $\mathcal{N}$ admits a feasible integer flow.*

Theorem 2. *Let G be an embedded bimodal digraph and let $\mathcal{N}$ be the associated flow network. For each feasible integer flow σ of $\mathcal{N}$ there is a quasi-upward planar drawing Γ of G, and for each quasi-upward planar drawing Γ of G there is a feasible integer flow σ of $\mathcal{N}$ such that:*

1. *The number of bends of Γ is equal to the cost of σ.*
2. *The number of bends in Γ along each boundary path between faces f and g is equal to $\chi(f, g)\sigma(f, g) + \chi(g, f)\sigma(g, f)$.*
3. *for each sink t, let f be the face such that $\sigma(t, f) = 1$; consider in Γ the horizontal line λ through t and a sufficiently small region R properly enclosing t; the intersection between R and the halfplane "above" λ belongs to f; and*
4. *for each source s, let f be the face such that $\sigma(s, f) = 1$; consider in Γ the horizontal line λ through s and a sufficiently small region R properly enclosing s; the intersection between R and the halfplane "below" λ belongs to f.*

Proof. Intuitively, k units of flow from face f to its adjacent face g through the arc (f, g) associated to boundary path p (shared by f and g) represent a $2k$-insert

switch operation performed on any edge e of p. The flow σ is interpreted as an assigment of sources and sinks to faces. The set of sources and sinks includes the new sources and sinks introduced by the insert switch operation. The $2k$ new sources and sinks introduced on e from the $2k$-insert-switch operation are assigned to face g. An analysis based on the effect of the insert switch operations on the capacity of the faces shows that the resulting assignment is upward consistent.

Theorem 2 allows to reduce the problem of finding a quasi-upward planar drawing of an n-vertex embedded bimodal digraph G with the minimum number of bends to a minimum cost flow problem. The flow problem can be solved in time $O(T(n))$, where $T(n)$ is equal to $O(n^2 logn)$, using a standard technique (see e.g. [1]). The $T(n)$ bound can be reduced to $O(n^{7/4}\sqrt{\log n})$ with a technique presented in [11].

If the digraph is not bimodal, a planarization technique can be used in a preprocessing step, where dummy vertices are introduced into the digraph to represent crossings.

Once the min-cost-flow problem has been solved a planar st-digraph including the given digraph can be found in time $O(n+b)$, where b is the number of bends, with a technique similar to the one presented in [3] to "saturate" the faces of an upward planar embedding. The enclosing planar st-digraph can be drawn with any of techniques for drawing planar st-digraphs (see [5]) and, eventually, the bends are smoothed.

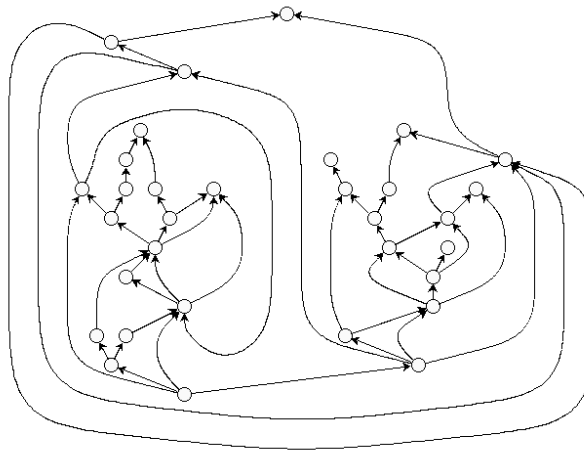


Fig. 4. A quasi-upward planar drawing

An example of drawing constructed with the algorithm presented in this section is shown in Fig. 4.

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

Let $G = (V, E)$ be a biconnected bimodal digraph and 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 along the edges of E' ($E' \subseteq E$). 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. We have:

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

Let $G'_{\phi'}$ be an embedded partial digraph of G with respect to the split components $G_1, \ldots, G_k$.

Let $\Gamma'_{\phi'}$ be a quasi-upward planar drawing of $G'_{\phi'}$. Suppose $\Gamma'_{\phi'}$ is such that $b_{E^{nonvirt}}(\Gamma'_{\phi'})$ is minimum. Consider an embedding ϕ of G that preserves ϕ' and an optimal quasi-upward planar drawing Γ_ϕ of G_ϕ.

Lemma 3. $b_{E^{nonvirt}}(\Gamma'_{\phi'}) \leq b_{E^{nonvirt}}(\Gamma_\phi)$.

Proof. Suppose, for a contradiction, that $b_{E^{nonvirt}}(\Gamma'_{\phi'}) > b_{E^{nonvirt}}(\Gamma_\phi)$. For each component G_i of G, the virtual path p_i is represented in Γ_ϕ by a polygonal line. We can derive from Γ_ϕ a quasi-upward planar drawing $\overline{\Gamma}_{\phi'}$ of $G'_{\phi'}$ by simply substituting the quasi-upward planar drawing of G_i with p_i, for each G_i. It is easy to see that $b_{E^{nonvirt}}(\overline{\Gamma}_{\phi'}) = b_{E^{nonvirt}}(\Gamma_\phi)$. Hence, $\overline{\Gamma}_{\phi'}$ has less bends along the edges that do not belong to virtual paths than $\Gamma'_{\phi'}$, a contradiction.

From Property 1 and Lemma 3 it follows a first lower bound.

Theorem 3. *Let $G = (V, E)$ be a biconnected bimodal digraph and $G'_{\phi'}$ an embedded partial graph of G. For each virtual path p_i of G', $i = 1, \ldots, k$, 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', consider any quasi-upward planar drawing Γ_ϕ of G_ϕ. We have that: $b(\Gamma_\phi) \geq b_{E^{nonvirt}}(\Gamma'_{\phi'}) + \sum_{i=1,\ldots,k} b_i$.*

A quasi-upward planar drawing $\Gamma'_{\phi'}$ of $G'_{\phi'}$ such that $b_{E^{nonvirt}}(\Gamma'_{\phi'})$ is minimum can be easily obtained by using the algorithm presented in Section 3. Namely, when two faces f and g share a virtual path, the corresponding edge of $\mathcal{N}$ in the minimum cost flow problem is set to zero.

A further lower bound is described in the following property.

Property 2. Let G_ϕ be an embedded bimodal digraph and $G'_{\phi'}$ an embedded partial digraph of G, such that ϕ preserves ϕ'. Consider an optimal quasi-upward planar drawing $\Gamma'_{\phi'}$ of $G'_{\phi'}$ and a quasi-upward planar drawing Γ_ϕ of G_ϕ. Then we have that: $b(\Gamma_\phi) \geq b(\Gamma'_{\phi'})$

The next theorem allows us to combine the above lower bounds into a hybrid technique.

Theorem 4. *Let G_ϕ be an embedded biconnected bimodal digraph and $G'_{\phi'}$ an embedded partial graph of G. 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 graph G_j of p_j. Consider a quasi-upward planar drawing $\Gamma'_{\phi'}$ of $G'_{\phi'}$, such that $b_{E^{nonvirt}}(\Gamma'_{\phi'}) + b_{E_F}(\Gamma'_{\phi'})$ is minimum. Let Γ_ϕ be a quasi-upward planar drawing of G_ϕ, we have that: $b(\Gamma_\phi) \geq b_{E^{nonvirt}}(\Gamma'_{\phi'}) + b_{E_F}(\Gamma'_{\phi'}) + \sum_{j:p_j \notin F^{virt}} b_j$*

5 Computing Optimal Drawings with Branch and Bound Techniques

Let G be a biconnected bimodal digraph. We describe a technique for enumerating all the possible quasi-upward planar drawings of G and strategies 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 [2] for orthogonal drawings.

The enumeration uses the SPQR-tree $\mathcal{T}$ of G. Namely, we enumerate all the quasi-upward planar drawings of G with edge e on the external face by rooting $\mathcal{T}$ at e and exploiting the capacity of $\mathcal{T}$ rooted at e in representing all the embeddings having e on the external face. A complete enumeration is done by rooting $\mathcal{T}$ at all the possible edges. 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 an "expansion" on all the vertices of G with more than one incoming or outgoing edge. The expansion replaces vertex v with its incoming edges $(u_1, v), \ldots, (u_h, v)$ with $h > 1$ and its outgoing edges $(v, w_1), \ldots, (v, w_k)$ with $k > 1$ with vertices v_1, v_2 and edges $(u_1, v_1), \ldots, (u_h, v_1), (v_1, v_2), (v_2, w_1), \ldots, (v_2, w_k)$. We call edge (v_1, v_2), that represents vertex v, *straight edge*. Now, the edges incident on v must enforce the bimodal constraint in any computed embedding. Observe that the straight edges are boundary paths.

We encode the embeddings of G as follows. We visit $\mathcal{T}$ in such a way that a node is visited after its parent, e.g. depth-first or breadth-first. This induces a numbering $1, \ldots, r$ 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 variable x_i of X that corresponds to an R-node μ_i can be set to three values corresponding to two swaps of the pertinent digraph of μ_i plus one unknown value. Each variable x_j of X that corresponds to a P-node μ_j with degree k (including the reference edge) can be set to up to $(k-1)!$ values corresponding to the possible permutations of the pertinent digraphs of the children of μ_j plus one unknown value. Unknown values represent portions of the embedding that are not yet specified.

A *search tree* $\mathcal{B}$ is defined as follows. Each node β of $\mathcal{B}$ corresponds to a different setting X_β of X. Such a setting is partitioned into two contiguous (one of them possibly empty) subsequences $x_1, \ldots, x_h$ and $x_{h+1}, \ldots, x_r$. Elements of the first subsequence contain values specifying embeddings, while elements in the second subsequence contain unknown values. The leaves of $\mathcal{B}$ are in corre-

spondence with settings of X with no unknown values. Internal nodes of $\mathcal{B}$ are in correspondence with settings of X with at least one unknown value. The setting of the root of $\mathcal{B}$ consists only of unknown values. The children of β (with subsequences $x_1, \ldots, x_h$ and $x_{h+1}, \ldots, x_r$) have subsequences $x_1, \ldots, x_{h+1}$ and $x_{h+2}, \ldots, x_r$, one child for each possible value of x_{h+1}.

Observe that there is a mapping between the embedded partial digraphs of G and the nodes of $\mathcal{B}$. Namely, the embedded partial digraph G_β of G associated to node β of $\mathcal{B}$ with subsets $x_1, \ldots, x_h$ and $x_{h+1}, \ldots, x_r$ is obtained as follows. First, set G_β to $skeleton(\mu_1)$, embedded according to x_1. Second, substitute each virtual path p_i of μ_1 with the skeleton of the child μ_i of μ_1, embedded according to x_i, only for $2 \leq i \leq h$. Then, recursively substitute virtual paths with embedded skeletons until all the skeletons in $\{skeleton(\mu_1), \ldots, skeleton(\mu_h)\}$ have been used.

We visit $\mathcal{B}$ breadth-first starting from the root. At each 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 planar drawing of G such that its embedding is (partially) specified by X_β. The current optimal solution is updated accordingly. The subtree rooted at β is not visited if the lower bound is greater than the current optimum.

For each β, lower bounds and upper bounds are computed as follows.

1. We construct G_β by using an array of pointers to the nodes of $\mathcal{T}$.
2. We compute lower bounds by using the results presented in Section 4.
   G_β is a partial digraph of G, with embedding derived from X_β. Let E^{virt} ($E^{nonvirt}$) be the set of edges (not) belonging to the virtual paths of G_β. For each virtual path p_i of G_β consider the pertinent digraph G_i of p_i and compute a lower bound b_i on the number of bends of G_i. 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.
   We apply the algorithm presented in Section 3 on G_β, assigning zero costs to the arcs of $\mathcal{N}$ associated to the virtual paths of G_β that are not in F^{virt}. In order to have no bends on the straight edges we set the cost of the corresponding arcs 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 minimum number of bends on the set $E^{nonvirt} \cup E_F$. Let $b(\Gamma_\beta)$ be such a number of bends. Then, by Theorem 4 we compute the lower bound L_β at node β, as $L_\beta = b(\Gamma_\beta) + \sum_{i:p_i \notin F^{virt}} b_i$.
   Lower bounds b_i can be pre-computed with a suitable pre-processing by visiting $\mathcal{T}$ bottom-up. The pre-computation consists of two phases. (a) We apply the algorithm presented in Section 3 on the skeletons of each R- and P-node, with zero cost for the arcs associated to the virtual paths; in this way we associate a lower bound to each R- and P-node of $\mathcal{T}$. Note that these pre-computed bounds do not depend on the choice of the reference edge, so they are computed only once, at the beginning of the computation. (b) We visit $\mathcal{T}$ bottom-up summing for each node μ the lower bounds of the children

of μ to the lower bound of μ. Note that these pre-computed bounds depend on the choice of the reference edge, so they are re-computed at any choice of the reference edge.

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 unknown values of X_β with embedding values in a random way. Then we apply the algorithm presented in Section 3 to G_ϕ so obtaining the upper bound. We also avoid multiple generations of the same embedded digraph in completing the partial digraph.

A further speed-up of the branch-and-bound technique can be obtained by suitably choosing the paths used as virtual paths in the skeletons of the nodes of $\mathcal{T}$. We call *switches* the vertices of a simple path where the direction of the edges changes. Observe that, in general, a boundary path with a few switches has less degrees of freedom in a quasi-upward planar drawing than a path with many switches. Hence, to have tighter lower bounds it is a good choice to select as virtual paths those that have a minimum number of switches. The feasibility of this approach is guaranteed by the following theorem.

Theorem 5. *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 computed in $O(n + m)$ time.*

Proof. An algorithm that works in $O(n+m)$ time computes a sequence of depth-first-search, "moving from u to v" and alternating in considering the direction of the edges.

6 Experimental Results

The algorithm presented in Section 3 has been implemented and extensively tested. It constitutes a fundamental "subroutine" for the implementation of the algorithm presented in Section 5 for biconnected digraphs. We have tested such branch and bound algorithm against a randomly generated test suite consisting of about 300 digraphs. The test suite is available on the Web and has been generated as follows (`www.dia.uniroma3.it/people/gdb/wp12/ LOG.html`). Any embedded planar biconnected graph can be generated from the triangle graph by means of a sequence of *insert-vertex* and *insert-edge* operations. Insert-vertex subdivides an existing edge into two new edges separated by a new vertex. Insert-edge inserts a new edge between two existing vertices that are on the same face. We have implemented a generation mechanism that at each step randomly chooses which operation to apply and where 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 graphs is in the range 1.2–2.

The implementation uses the C++ language and the `GDToolkit` library (`www.dia.uniroma3.it/people /gdb/wp12`). Experiments have been done with

a Sun Ultrasparc 1. The results of the experiments are summarized in the graphics of Fig. 5. To check the applicability of the algorithm we have measured the CPU time (Fig. 5.a). To better understand the features of the test suite we have measured the number of embeddings (Fig. 5.b) and the number of components affecting the computation time (Fig. 5.c).

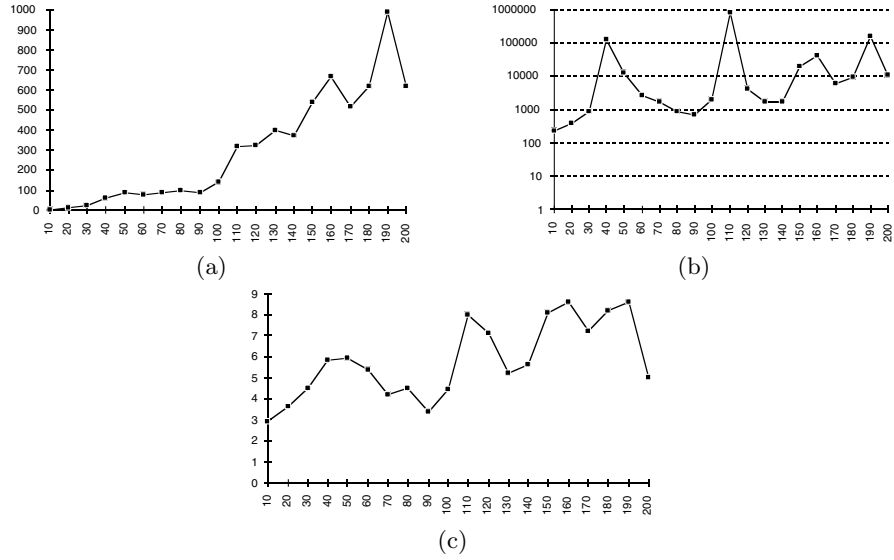


Fig. 5. Graphics summarizing the experiments: (a) CPU time (seconds), (b) number of embeddings (log. scale), and (c) number of components (sum of P and R nodes). The x-axis represents the number of vertices and in the y-axis we give average values.

All the figures presented in this paper have been drawn with the `GDToolkit` system with an implementation of the algorithms of Sections 3 and 5.

7 Conclusions and Open Problems

We have 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. [19, 20]) 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 [5]) 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 the number of bends of each biconnected component by searching all the possible topologies of the component. We have also shown that the algorithm has a reasonable time performance for digraphs 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 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?

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 number of bends. In *Proc. 5th Workshop Algorithms Data Struct.*, volume 1272 of *Lecture Notes Comput. Sci.*, pages 331–344. Springer-Verlag, 1997.

[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. In *Proc. 1st Annu. European Sympos. Algorithms*, volume 726 of *Lecture Notes Comput. Sci.*, pages 37–48. Springer-Verlag, 1993.

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

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

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

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

[9] 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.

[10] A. Garg and R. Tamassia. Upward planarity testing. *Order*, 12:109–133, 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)*, Lecture Notes Comput. Sci. Springer-Verlag, 1997.

[12] J. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2:135–158, 1973.

[13] M. D. Hutton and A. Lubiw. Upward planar drawing of single-source acyclic digraphs. *SIAM J. Comput.*, 25(2):291–311, 1996.

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

[15] X. Lin and P. Eades. Area requirements for drawing hierarchically planar graphs. In G. Di Battista, editor, *Graph Drawing (Proc. GD' 97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 219–229. Springer-Verlag, 1998.

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

[17] A. Papakostas. Upward planarity testing of outerplanar dags. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes Comput. Sci.*, pages 298–306. Springer-Verlag, 1995.

[18] E. Reingold and J. Tilford. Tidier drawing of trees. *IEEE Trans. Softw. Eng.*, SE-7(2):223–228, 1981.

[19] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.

[20] R. Tamassia, G. Di Battista, and C. Batini. Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst. Man Cybern.*, SMC-18(1):61–79, 1988.