

A Fully Dynamic Algorithm to Test the Upward Planarity of Single-Source Embedded Digraphs

Aimal Rextin and Patrick Healy

Computer Science Department, University of Limerick, Ireland
{aimal.tariq,patrick.healy}@ul.ie

Abstract. In this paper, we present a dynamic algorithm that checks if a single-source embedded digraph is upward planar in the presence of edge insertions and edge deletions. Let G_ϕ be an upward planar single-source embedded digraph and let $G'_{\phi'}$ be a single-source embedded digraph obtained by updating G_ϕ . We show that the upward planarity of $G'_{\phi'}$ can be checked in $O(\log n)$ amortized time when the external face is fixed.

1 Introduction

Assume we have a solution of a graph theoretic problem P on a graph G . A *dynamic graph algorithm* tries to solve P after G is updated in less time than recomputing P from scratch [5]. Dynamic graph algorithms are useful when a graph has discrete changes like the addition or deletion of vertices or edges. A practical example of a dynamic graph algorithm is the maintenance shortest paths in a communication network as links are added or deleted.

In this paper, we present a dynamic algorithm to check if a single-source embedded digraph remains upward planar after an edge is inserted or deleted. An *planar embedding* is an equivalence class of planar drawings for a graph G , such that each drawing of this class has the same circular order of edges around each vertex of G . A graph G with a given planar embedding is denoted by G_ϕ and we call it an *embedded digraph*. A digraph G is *upward planar* if it has a planar drawing with all edges pointing monotonically upward [6]. It is NP-hard to test if a digraph G is upward planar [9], hence upward planarity testing is either done for a fixed embedding [3,7], or for special classes of digraphs like single-source digraphs [10,4], series-parallel digraphs [8], and outer planar digraphs [11].

Let G_ϕ be an upward planar embedded digraph with the single-source s_G . We let $G'_{\phi'}$ be an embedded digraph with the single-source $s_{G'}$, such that $G'_{\phi'}$ is obtained from G_ϕ by performing one of the following update operations:

- **insert-edge**(e, u, v): Insert an edge $e = (u, v)$ between two existing vertices in G_ϕ .
- **attach-vertex**(e, u, v): Add a new vertex and insert an edge between an existing vertex and the new vertex.
- **delete-edge**(e): Delete the edge e from G_ϕ . We also delete a vertex if it results in no incident edge.

An update operation is illegal if the resulting digraph is not single-source. In case of an edge insertion, this happens if an edge $e = (u, v)$ is inserted between existing vertices such that $v = s_G$, or when edge $e = (u, v)$ is inserted between a new vertex u and $v \neq s_G$. An edge deletion is illegal if an edge $e = (u, v)$ is deleted such that $G'_{\phi'}$ becomes disconnected.

It is generally believed that upward planar drawings of a digraph are more comprehensible to humans. Hence, it is reasonable to say that a non-upward planar digraph H is more readable if the largest possible subgraph of H is drawn in an upward planar fashion. In this paper, we present a dynamic algorithm to test the upward planarity of $G'_{\phi'}$. Our dynamic algorithm can be used to compute a maximal upward planar subgraph for a single-source digraph H by incrementally building an upward planar embedding of H and discarding a new edge if it results in a non-upward planar embedded digraph.

In the remainder of this section, we define some basic terminology and review some relevant results. In Sec. 2, we discuss how to obtain a bimodal and embedded $G'_{\phi'}$. In Sec. 3, we give a characterization of upward planarity of $G'_{\phi'}$ with respect to the update operations. In Sec. 4, we present our algorithm and its complexity analysis. We conclude by identifying some related open problems.

1.1 Preliminaries

We assume basic familiarity with graph theory. Let G be a graph. We denote the set of vertices of G by $V(G)$ and we denote the set of edges of G by $E(G)$. In a digraph, a *source* vertex has only outgoing edges, a *sink* vertex has only incoming edges, and an *internal* vertex has both incoming and outgoing edges. A planar drawing Γ divides the plane into non-overlapping regions called *faces*; the unique unbounded region is called the *external face* and each bounded region is called an *internal face*. The *facial boundary* of a face f is the path enclosing f in the clockwise direction, all drawings of an embedded graph have the same set of facial boundaries. An embedded digraph G_{ϕ} is *bimodal* when $\phi(v)$ can be partitioned into two sets of consecutive incoming and outgoing edges for every vertex $v \in G_{\phi}$.

In an embedded digraph G_{ϕ} , an *angle* is a triplet $\langle e_1, v, e_2 \rangle$ such that the edges are incident to the vertex v and edge e_1 is immediately before edge e_2 in $\phi(v)$. A vertex v is incident to the angle $\langle e, v, e \rangle$ when e is the only edge incident to v . A *switch* $\langle e_1, v, e_2 \rangle$ is an angle with both e_1 and e_2 pointing either toward or away from v : it is a *sink-switch* when e_1 and e_2 point toward v and it is a *source-switch* when e_1 and e_2 point away from v [7]. Switches were originally defined as nodes in an embedded biconnected digraph by Bertolazzi *et al.* [3], however Didimo generalized their concept to general embedded digraphs by defining them as angles [7].

We now show that both G_{ϕ} and $G'_{\phi'}$ have at most one sink-switch incident to a vertex v inside a particular face. This allows us to refer to a vertex v incident to a sink-switch $\langle e_1, v, e_2 \rangle$ in a face f as *sink-switch v incident to face f* for simplicity and clarity.

Lemma 1. *Let G_{ϕ} be an upward planar embedded digraph with a single source s_G , and let $G'_{\phi'}$ be the bimodal embedded digraph with a single source $s_{G'}$ obtained after adding an edge in G_{ϕ} . Both G_{ϕ} and $G'_{\phi'}$ have at most one sink-switch incident to a vertex v inside a particular face.*

The *face-sink graph* F of G_ϕ is an undirected graph such that the vertices of F are the faces of G_ϕ and all vertices of G_ϕ that are incident to a sink-switch; an edge (f, v) is in F if face f is incident to a sink-switch on a vertex v in G_ϕ . Bertolazzi *et al.* [4] presented an $O(n)$ -time algorithm to test the upward planarity of a single-source embedded digraph G_ϕ . This algorithm is based on the following theorem:

Theorem 1 (Bertolazzi *et al.* [4]). *Let G_ϕ be an embedded digraph with a single-source s_G . G_ϕ is upward planar with face h as the external face if and only if the following conditions are satisfied.*

1. *The face-sink graph F of G_ϕ is a forest.*
2. *F has exactly one tree \hat{T} with no internal vertices, while all other trees have exactly one internal vertex.*
3. *\hat{T} contains the node corresponding to face h and s_G is incident to face h in G_ϕ .*

2 Maintaining Planarity and Bimodality

Theorem 1 requires an embedded single-source digraph, however bimodality is a necessary condition for upward planarity and hence $G'_{\phi'}$ will have more chances to be upward planar if it is already bimodal and planar. In this section, we see how a bimodal embedded digraph $G'_{\phi'}$ can be obtained after G_ϕ is updated. The embedded digraph G_ϕ will remain bimodal and planar after an edge is deleted, hence we only study the case when an edge is inserted.

When an edge is inserted, a planar and bimodal embedded digraph $G'_{\phi'}$ can be obtained, if it exists, by using the techniques of Bertolazzi *et al.* [2] and Tamassia [12]. Tamassia described a technique to incrementally build a planar embedding: it checks if an edge can be added to the current embedded graph without introducing a crossing in $O(\log n)$ time and it then adds the new edge to the current embedded graph in $O(\log n)$ amortized time [12]. A technique for constructing a bimodal embedding of a digraph \mathcal{G} was discussed by Bertolazzi *et al.* [2]. It works by splitting all vertices of \mathcal{G} with at least 2 incoming edges and at least 2 outgoing edges into a vertex v_a with all the incoming edges of v and a vertex v_b with all the outgoing edges of v , and adding the edge (v_a, v_b) . We call the vertices that are split as *split-vertices* and we call the resulting digraph as the *split-digraph* $\tilde{\mathcal{G}}$. Bertolazzi *et al.* showed that \mathcal{G} has a planar bimodal embedding if and only if $\tilde{\mathcal{G}}$ has a planar embedding. We get a planar and bimodal embedded \mathcal{G}_ϕ by merging back the split vertices in a planarly embedded $\tilde{\mathcal{G}}_\phi$ [2].

We obtain a bimodal and planarly embedded $G'_{\phi'}$, if it exists, by maintaining a corresponding planarly embedded split-digraph \tilde{G}_ϕ . Figure 1 shows an embedded digraph G_ϕ and its corresponding split-digraph \tilde{G}_ϕ . A vertex v in G has two corresponding vertices \tilde{v}_a and \tilde{v}_b in \tilde{G} if it is a split-vertex, and it has one corresponding vertex \tilde{v} otherwise. If v is a split-vertex, we let \tilde{v}_b represent the vertex in \tilde{G} with all corresponding outgoing edges of v , and we let \tilde{v}_a represent the vertex in \tilde{G} with all corresponding incoming edges of v . We define a function $o : V(G) \rightarrow V(\tilde{G})$, such that $o(v) = \tilde{v}_b$ when v is a split-vertex and $o(v) = \tilde{v}$ otherwise. Similarly, we define function $i : V(G) \rightarrow V(\tilde{G})$, such that $i(v) = \tilde{v}_a$ when v is a split-vertex and $i(v) = \tilde{v}$ otherwise. We also define a function

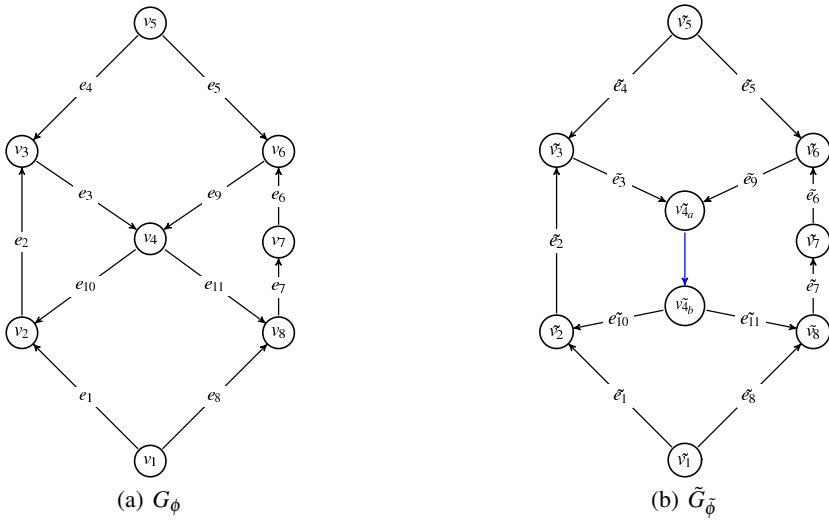


Fig. 1. An embedded digraph G_ϕ (a); and its embedded split digraph \tilde{G}_ϕ (b)

$e : E(G) \rightarrow E(\tilde{G})$, which maps the edges in G to their corresponding edges in \tilde{G} . When we want to add an edge $e = (u, v)$ in G_ϕ , we first try to add the edge $\tilde{e} = (o(u), i(v))$ in \tilde{G}_ϕ . The embedded digraph G'_ϕ is not planar or bimodal if \tilde{e} cannot be added in \tilde{G}_ϕ using Tamassia’s method. Lets assume that we get a planar \tilde{G}'_ϕ , with \tilde{e} inserted between \tilde{e}_1 and \tilde{e}_2 at $o(u)$, and \tilde{e} inserted between \tilde{e}'_1 and \tilde{e}'_2 at $i(u)$. In this case, we get a planar and bimodal G'_ϕ by adding e between $e^{-1}(\tilde{e}_1)$ and $e^{-1}(\tilde{e}_2)$ at u , and adding e between $e^{-1}(\tilde{e}'_1)$ and $e^{-1}(\tilde{e}'_2)$ at v . Figure 1 shows that we can bimodally add the edge (v_5, v_4) in G_ϕ but not the edge (v_4, v_5) .

The split-digraph \tilde{G}_ϕ takes $O(n)$ space. If the addition of edge e makes a vertex v a split-vertex then we will need to construct the corresponding \tilde{v}_a and \tilde{v}_b in \tilde{G}_ϕ . This can be done in constant time because there will be either one incoming edge or one outgoing edge incident to v before the new edge is added. Hence we have the following lemma.

Lemma 2. *Let G_ϕ be an upward planar embedded digraph and let e be an edge that we want to insert in G_ϕ . We can perform the following two operations.*

1. *Check if an edge e can be added to G_ϕ such that the resulting graph has a bimodal and planar embedding in $O(\log n)$ time.*
2. *If the previous test is true then we can obtain a planar and bimodal embedded digraph G'_ϕ in $O(\log n)$ amortized time.*

The insertion of an edge $e = (u, v)$ bisects an angle $\alpha_u = \langle e_1, u, e_2 \rangle$ at vertex u into two new angles $\langle e_1, u, e \rangle$ and $\langle e, u, e_2 \rangle$. The new edge e similarly bisects the angle α_v at vertex v into two new angles. The insert-face f is divided into two new faces

f_1 and f_2 when a new edge $e = (u, v)$ is inserted when both u and v already exist. Let the facial boundary of f be $w_0, e_0, \dots, e_i, u, e_{i'}, \dots, e_j, v, e_{j'}, \dots, e_k, w_k = w_0$. After e is inserted, let $\langle e_i, u, e_{i'} \rangle$ and $\langle e_j, v, e_{j'} \rangle$ be the angles that are bisected at u and v respectively, then f_1 has the facial boundary $e, u, e_{i'}, \dots, e_j, v, e$ and f_2 has the boundary $w_0, e_0, \dots, u, e, v, \dots, e_k, w_k = w_0$. If both u and v exist and α_v is not a switch then either f_1 or f_2 will have a sink-switch at v . We assume, without the loss of generality, that the new sink-switch will be created at f_1 . Similarly, when a new edge $e = (u, v)$ is inserted in f and one of the vertices is new, then the facial boundary of f will change. Let the facial boundary of f be $w_0, e_0, \dots, e_i, w', e_{i'}, \dots, e_k, w_k = w_0$, such that w' is the existing vertex and $\langle e_i, w', e_{i'} \rangle$ is the angle bisected at w' . After e is inserted, the facial boundary will change to $w_0, e_0, \dots, e_i, w', e, w'', e, w', \dots, e_k, w_k = w_0$, where w'' is the new vertex. Hence, we can maintain the facial boundaries in a linked list which can be updated in constant time by keeping pointers to nodes in the linked list.

3 Maintaining Upward Planarity

In this section, we characterize the upward planarity of $G'_{\phi'}$ after an update operation. We will only study the case of inserting an edge because $G'_{\phi'}$ remains upward planar when an edge is deleted. We will however need to update our datastructures when an edge is deleted, this is discussed in the next section. We assume that $G'_{\phi'}$ is bimodal and planar because we construct it by using the method described in Sec. 2.

Let F be the face-sink graph corresponding to G_{ϕ} and let F' be the face-sink graph corresponding to $G'_{\phi'}$. Since G_{ϕ} is upward planar, F will satisfy Theorem 1. Further, $G'_{\phi'}$ will be upward planar if and only if F' satisfies Theorem 1. In this section, we show that we can check if F' satisfies Theorem 1 by considering a small subset of F' . This will lead to an efficient dynamic single-source upward planarity testing algorithm, which is presented in the next section.

We first present some definitions that will be used later in this section. An edge e is inserted in one particular face of G_{ϕ} , which we call the *insert-face* and denote it by f . Every face g in G_{ϕ} has a corresponding vertex \bar{g} in F . Let $T_f = (V_{T_f}, E_{T_f})$ be the tree that contains \bar{f} , i.e. the vertex corresponding to f . Let T be a tree in F , we define $faces(T)$ to be the set of faces such that a face g is in $faces(T)$ if and only if $\bar{g} \in V(T)$. We also define a set of vertices, denoted by $nodes(T)$, that contains all vertices of G_{ϕ} that are in $V(T)$. Let \hat{T} denote the tree of F with no internal vertices, and let $H_{G_{\phi}}$ denote the set of faces in G_{ϕ} that are incident to the single-source s_G . Then $H_{G_{\phi}} \cap faces(\hat{T})$ is the set of all possible external faces in an upward planar drawing of G_{ϕ} .

Our results in this section rely on observing how F changes into F' . We define a tree T_1 to be different from a tree T_2 if it has at least one different vertex or one different edge. We claim that either $F \setminus F' = \{T_f\}$ or $F \setminus F' = \emptyset$. This is because a tree T in F will transform to a new tree T' in F' only if a new sink-switch is added in a face $g \in faces(T)$ or a sink-switch is removed from g or when g is divided into two new faces. This can happen only for the insert-face f , hence at most T_f will be transformed by the edge insertion.

We now have a closer look at the structure of T_f . If we traverse f in the clockwise direction, we will encounter some vertices that are incident to a sink-switch in f . Let

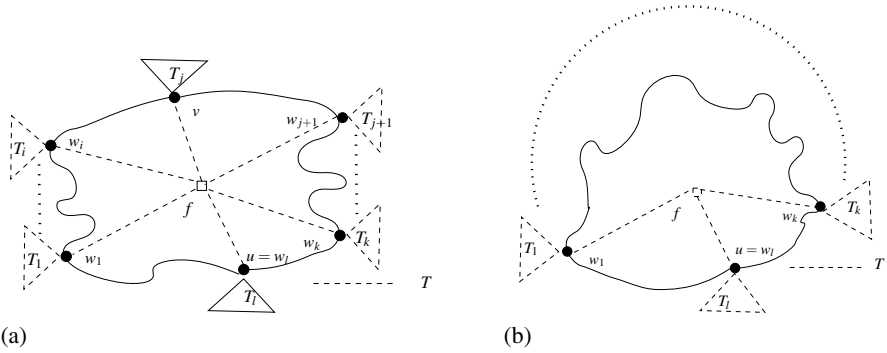


Fig. 2. The tree T_f with respect to face f such that both u and v already exist (a); and the tree T_f with respect to face f when only u is the existing vertex (b)

$W = \{w_1, w_2, \dots, w_l\}$ be all such vertices. We define a subtree T_i as the part of T_f that is reachable from \bar{f} through the vertex w_i , where $1 \leq i \leq l$. We call $w_i \in T_i$ the *access-vertex* of T_i with respect to \bar{f} . This is shown in Fig. 2 for both type of edge insertions discussed in Sec. 1. When both end vertices of $e = (u, v)$ already exist in G_ϕ then the partitioning of T_f is shown in Fig. 2(a): the access-vertex for T_1, \dots, T_i is between u and v in the clockwise direction, the access-vertex of T_{j+1}, \dots, T_k is between v and u in the clockwise direction, the access-vertex of T_j is v , and the access-vertex of T_l is u . Note that T_j and T_l will be empty subgraphs if u and v are not incident to a sink-switch in f . When $e = (u, v)$ has one existing vertex u then the partitioning is shown in Fig. 2(b): the access-vertex at u (if it exists) is T_l , and the access-vertices for T_1, \dots, T_k are encountered as we traverse f in the clockwise direction after u .

The next lemma is easily derived from the illegal operations described in Sec. 1 and the fact that $G'_{\phi'}$ is bimodal.

Lemma 3. *If edge $e = (u, v)$ is added in G_ϕ such that both u and v already exist, then α_v , the bisected angle at v , cannot be a source-switch.*

Proof. Assume that v is incident to a source-switch in f . We know from Sec. 1 that $v \neq s_G$ hence v has at least one incoming edge. This implies that $G'_{\phi'}$ is not bimodal, which is a contradiction. \square

We now come to the main results of this section, presented as a series of theorems. We divide the analysis into two main cases: $T_f \neq \hat{T}$ and $T_f = \hat{T}$. When $T_f \neq \hat{T}$, the tree \hat{T} is in F' and all other trees in $F' \cap F$ have one internal vertex. In this case, $G'_{\phi'}$ will be upward planar if all trees of $F' \setminus F$ have one internal vertex. The single internal vertex of T_f is denoted by w_{T_f} when $T_f \neq \hat{T}$. On the other hand, all trees in $F' \cap F$ have one internal vertex when $T_f = \hat{T}$. In this case, $G'_{\phi'}$ will be upward planar if $F' \setminus F$ has one tree T with no internal vertex, all other trees trees in $F' \setminus F$ have exactly one internal vertex and $faces(T) \cap H_{G'_{\phi'}} \neq \emptyset$.

In some cases, $G'_{\phi'}$ will always be upward planar, the next theorem analyze these cases.

Theorem 2. *Let G_{ϕ} be an upward planar embedded digraph with a single source s_G and a face-sink graph F . If we insert an edge $e = (u, v)$ in the face $f \in G_{\phi}$, then $G'_{\phi'}$ will be upward planar if one of the following conditions is true.*

1. Both u and v already exist, such that α_v is a sink-switch and α_u is either a source-switch or α_u is a non-switch angle;
2. u is the new vertex;
3. v is the new vertex and α_u is either a source-switch or a non-switch angle.

We analyze the remaining cases by looking at the different possibilities for α_u and α_v . Both α_u and α_v can either be a sink-switch, a source-switch, or they can be a non-switch angle. If the new edge is added between two existing vertices, then we know from Lemma 3 that α_v cannot be a source-switch. The case when α_v is a sink-switch and α_u is either a source-switch or when α_v is a non-switch angle is already discussed in Theorem 2. Hence we need to analyze when α_v is a sink-switch or a non-switch angle, while α_u is any type of angle. These cases are discussed in Theorem 3 and Theorem 4. Theorem 3 discusses the case when α_v is not a switch while α_u can be any type of angle. The only case left for both end-vertices to be already existing is when both α_v and α_u are sink-switches, which is discussed in Theorem 4.

Theorem 3. *Let G_{ϕ} be an upward planar embedded digraph with a single source s_G and a face-sink graph F . If we insert an edge $e = (u, v)$ in the face $f \in G_{\phi}$ such that both u and v exists and α_v is not a switch then $G'_{\phi'}$ will be upward planar if and only if one of the following is true.*

1. If $T_f \neq \hat{T}$ then $w_{T_f} \in \text{nodes}(T_{j+1} \cup \dots \cup T_k)$.
2. If $T_f = \hat{T}$ then $s_{G'}$ is incident to at least one face in $\text{faces}(T_{j+1} \cup \dots \cup T_k)$.

While α_u can either be a sink-switch, or, a source-switch, or not a switch.

Proof. The tree T_f is transformed into two new trees \mathcal{T}_1 and \mathcal{T}_2 , such that

$$\begin{aligned}
 V(\mathcal{T}_1) &= V(T_1 \cup T_2 \dots T_j) \\
 E(\mathcal{T}_1) &= E(T_1 \cup T_2, \dots T_j) \cup \{(\bar{f}_1, w_1) \cup \dots (\bar{f}_1, w_j)\} \\
 V(\mathcal{T}_2) &= V(T_{j+1} \cup T_{j+2} \dots T_k) \\
 E(\mathcal{T}_2) &= E(T_{j+1} \cup T_{j+2} \dots T_k) \cup \{(\bar{f}_2, w_{j+1}) \cup \dots (\bar{f}_2, w_k)\}
 \end{aligned}$$

where w_i is the access node for subtree T_i . When u is a sink-switch in f , there will also be a third tree

$$\mathcal{T}_3 = (V(T_i), E(T_i)).$$

We may observe that v is an internal vertex that is part of \mathcal{T}_1 and u is an internal vertex of \mathcal{T}_3 (when \mathcal{T}_3 exists). All possible cases are shown in Fig. 3.

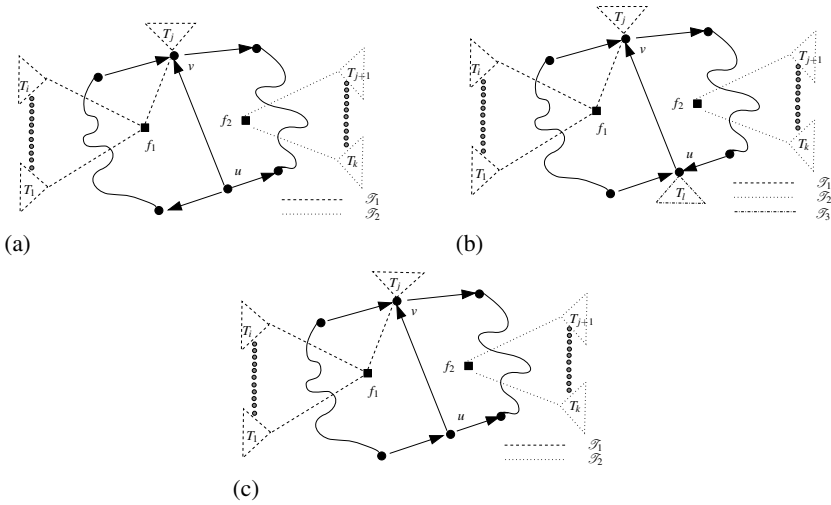


Fig. 3. Cases of Theorem 3: u and v already exist, α_u is source-switch and α_v is not a switch (a); u and v already exist, α_u is not a switch and α_v is sink-switch (b); u and v already exist, α_u is a sink-switch and α_v is not a switch (c)

If: If $T_f \neq \hat{T}$ and $w_{T_f} \in \text{nodes}(T_{j+1} \dots \cup T_k)$ then each tree in $F' \setminus F$ has one internal vertex and $G'_{\phi'}$ is upward planar in this case. Similarly, if $T_f = \hat{T}$ then each of the generated new trees has one internal vertex except \mathcal{T}_2 . Again, $G'_{\phi'}$ is upward planar because according to our assumption $\text{faces}(\mathcal{T}_2) \cap H_{G'_{\phi'}} \neq \emptyset$.

Only if: We show this by proving the contrapositive. If $T_f \neq \hat{T}$ and $w_{T_f} \in \{T_1, \dots, T_j\}$ then F' has two trees \mathcal{T}_2 and \hat{T} that have no internal vertices. Similarly, if $T_f = \hat{T}$ and $s_{G'}$ is not incident to a face in $\text{faces}(T_{j+1} \dots \cup T_k)$ then \mathcal{T}_2 has no internal vertex but $\text{faces}(\mathcal{T}_2) \cap H_{G'_{\phi'}} = \emptyset$. Hence $G'_{\phi'}$ will not be upward planar. \square

Theorem 4. Let G_{ϕ} be an upward planar embedded digraph with a single source s_G and a face-sink graph F . We insert an edge $e = (u, v)$ in the face $f \in G_{\phi}$, such that both u and v already exists. If both α_u and α_v are sink-switches in f then $G'_{\phi'}$ will be upward planar if and only if one of the following is true.

1. If $T_f \neq \hat{T}$ then $w_{T_f} \in \text{nodes}(T_1 \cup \dots \cup T_k)$.
2. If $T_f = \hat{T}$ then $s_{G'}$ is incident to at least one face in $\text{faces}(T_1 \cup \dots \cup T_k)$

When one of the end-vertices is a new vertices for the new edge $e = (u, v)$, then the case when u is the new vertex and when v is the new vertex and α_u is a sink-switch is already discussed in Theorem 2. The only remaining case is discussed in Theorem 5.

Theorem 5. Let G_{ϕ} be an upward planar embedded digraph with a single source s_G and a face-sink graph F . We insert an edge $e = (u, v)$ in the face $f \in G_{\phi}$, such that v is a new vertex and α_u is a sink-switch, then $G'_{\phi'}$ will be upward planar if and only if one of the following is true.

1. If $T_f \neq \hat{T}$ then $w_{T_f} \in \text{nodes}(T_1 \cup \dots \cup T_k)$.
2. If $T_f = \hat{T}$ then $s_{G'}$ is incident to at least one face in $\text{faces}(T_1 \cup \dots \cup T_k)$.

4 Algorithm and Time Complexity

We now present our algorithm for testing the upward planarity of a bimodal and planarly embedded $G'_{\phi'}$ with a fixed external face and discuss its complexity. The input to the algorithm is G_{ϕ} , the upward planar embedded digraph; e , the edge to be added or deleted; and $\tilde{G}_{\tilde{\phi}}$, the embedded split-digraph corresponding to G_{ϕ} . It first constructs a bimodal and planar $G'_{\phi'}$, if it exists. If we delete the edge e then the resulting $G'_{\phi'}$ will also be upward planar. The rest of the algorithm checks if $G'_{\phi'}$ satisfies the conditions of theorems from the previous section. The algorithm is shown in Algorithm 1.

We now show that dynamic upward planarity testing based on Theorem 1 requires $\Omega(n)$ time when we allow the external face to change and do not transform G_{ϕ} . We show it by assuming that $G'_{\phi'}$ is non-upward planar with h as its external face, where h was the external face of G_{ϕ} . The digraph $G'_{\phi'}$ will be upward planar if there is a face $g \neq h$, such that $g \in H_{G'_{\phi'}} \cap \text{faces}(\hat{T}')$, where \hat{T}' is the tree in F' with no internal vertex. Recall that $H_{G'_{\phi'}}$ denotes the set of faces that are incident to the single-source in $G'_{\phi'}$. In order to find an alternative external face g in $o(n)$ time, we dynamically maintain $H_{G'_{\phi'}}$ by making appropriate additions or deletions in $H_{G_{\phi}}$ because recomputing $H_{G'_{\phi'}}$ from scratch will take $O(n)$ time. Now, if the new edge $e = (u, v)$ is between a new vertex u and an existing vertex $v = s_G$, then $H_{G'_{\phi'}} = \{f\}$. This results in a contradiction because removing the old faces will take $O(n)$ time. Hence it is not possible to design an efficient dynamic upward planarity testing algorithm for single-source embedded digraphs using Theorem 1.

We recall from Sec. 2 that finding a planar and bimodal $G'_{\phi'}$ requires $O(\log n)$ amortized time. We can check that an insertion satisfies Theorem 2 in constant time. Let μ represent the unique internal vertex w_{T_f} of T_f when $T_f \neq \hat{T}$ and represent the external face h of G_{ϕ} when $T_f = \hat{T}$. The overall time complexity of Algorithm 1 depends on how efficiently we can check if μ is in a particular subtree of T_f . The location of μ can be easily be determined in $O(n)$ time by traversing the nodes of T_f , but then the time complexity of Algorithm 1 will equal running the algorithm of Bertolazzi *et al.* from scratch. We propose instead an $O(1)$ -time method. We maintain a directed version of F by rooting each tree $T \in F$ at its unique internal vertex or vertex corresponding to the external face, and then orienting all edges toward the root. Each vertex $v \neq \mu$ will have exactly one outgoing edge and if $v = \mu$ then it has no outgoing edge. Let $out(v)$ represent the outgoing edge for a vertex v and let $p(v)$ be the target node for $out(v)$. Note that, $p(\bar{f})$ is always an access-vertex w_i for a subtree T_i .

We can check if μ is in a subtree satisfying Theorems 3, 4 or 5 by finding the relative location of $p(\bar{f})$ in the facial boundary of f . This is done by maintaining a linked list L_f for every face $f \in G_{\phi}$, such that every vertex $v \in f$ has a corresponding real number $L_f[v]$. We construct $L_f = \{L_f[v_1], \dots, L_f[v_k]\}$ such that: $L_f[v_i] < L_f[v_{i+1}]$, where v_1, \dots, v_k are consecutive vertices on the facial boundary of f in the clockwise direction.

Algorithm 1. Dynamic Upward Planarity Test ($G_\phi, e = (u, v), \tilde{G}_{\tilde{\phi}}$)

```

1: Find a planar and bimodal embedding of  $G', G'_\phi$ 
2: if we cannot find a planar and bimodal  $G'_\phi$  then
3:   Return False
4: end if
5: if delete the edge  $e$  then
6:   Return True
7: end if
8: if Both  $u$  and  $v$  already exist and  $\alpha_v$  is sink-switch; or  $u$  is a new vertex; or  $v$  is a new vertex
   and  $\alpha_u$  is not sink-switch then
9:   Return True
10: end if
11:  $\mu = w_{T_f}$  when  $T_f \neq \hat{T}$  and  $\mu = h$  when  $T_f = \hat{T}$ 
12: if  $u$  and  $v$  exist and  $\alpha_v$  is not a switch and  $\mu \in \{T_{j+1}, \dots, T_k\}$  then
13:   Return True
14: else if  $u$  and  $v$  exist and both  $\alpha_u$  and  $\alpha_v$  are sink-switches and  $\mu \in \{T_1, \dots, T_k\}$  then
15:   Return True
16: else if  $v$  is a new vertex and  $\alpha_u$  is a sink-switch and  $\mu \in \{T_1, \dots, T_k\}$  then
17:   Return True
18: else
19:   Return False
20: end if

```

When the insertion of $e = (u, v)$ divides f into f_1 and f_2 , we divide L_f to get L_{f_1} and L_{f_2} such that $L[u]$ and $L[v]$ are present in both of them. We also maintain pointers from each vertex incident to a face f to its entry in L_f . When we insert an edge e such that a new vertex v_j is added, we can choose a sufficiently small ε , letting $L_f[v_j] = L_f[v_{j-1}] + \varepsilon$. However, this can result in difficulties associated with high precision real numbers and hence increase the time complexity in comparing two elements of L_f . Instead, we suggest using the algorithm by Bender *et al.* to assign $L_f[v_j]$ in $O(\log n)$ amortized time [1]. The algorithm by Bender *et al.* maintains a dynamic list and allows a user to compare the order of any two elements in the list. This is done by assigning tags of $O(\log n)$ bits to each element in the list. Hence, any $L_f[v_j]$ and $L_f[v_i]$ can be efficiently compared. The following lemma shows that L_f can be used to efficiently check if μ is in the required subtree. This technique will also work when L_f is divided into two new lists L_{f_1} and L_{f_2} because the algorithm of Bender *et al.* assigns a tag by locally relabeling a subset of a list.

Lemma 4. *We can check the conditions of Theorems 3, 4 5 in constant time.*

Proof. Theorem 4 and 5: We need to check if $p(\bar{f}) \in \{w_1, \dots, w_k\}$, this will be true if $p(\bar{f}) \neq w_l$. Hence $p(\bar{f}) \in \{w_1, \dots, w_k\}$ and G'_ϕ will be upward planar if and only if $L_f[p(\bar{f})] \neq L_f[u]$.

Theorem 3: We can see from Fig. 3 that we need to check if $p(\bar{f}) \in \{w_{j+1}, \dots, w_k\}$. We have the following 2 cases, based on the fact that $p(\bar{f})$ should be between v and u in the clockwise direction in order to satisfy the theorem.

1. When $L_f[v] < L_f[u]$ then $p(\bar{f}) \in \{w_{j+1}, \dots, w_k\}$ will be true if $L_f[v] < L_f[p(\bar{f})] < L_f[u]$.
2. When $L_f[u] < L_f[v]$ then $p(\bar{f}) \in \{w_{j+1}, \dots, w_k\}$ will be true if either $L_f[p(\bar{f})] < L_f[u] < L_f[v]$ or $L_f[u] < L_f[v] < L_f[p(\bar{f})]$.

□

We have yet to show that we maintain the correct orientation of the edges of F in the presence of updates. The following two lemmas shows that we can do this in constant time. We define the *splitting* of a vertex \bar{f} with respect to the new edge $e = (u, v)$ as the creation of two new vertices \bar{f}_1 and \bar{f}_2 , such that \bar{f}_1 has edges of \bar{f} to and from w_1, \dots, w_i and \bar{f}_2 has all edges of \bar{f} to and from w_{j+1}, \dots, w_k . We also define the *merging* of a vertex \bar{f}_1 and a vertex \bar{f}_2 as the creation of a new vertex \bar{f} , such that \bar{f} has all outgoing edges and incoming edges of both \bar{f}_1 and \bar{f}_2 . We need to split \bar{f} when as a result of edge insertion the face f splits into f_1 and f_2 , and we need merging when two faces f_1 and f_2 combine to form the face f . Splitting and merging \bar{f} can be done by splitting and merging the adjacency list of \bar{f} .

Lemma 5. *Let G_ϕ be an upward planar embedded digraph with a single-source s_G . If we add a new edge e to create an embedded digraph G'_ϕ with a single source $s_{G'}$ such that G'_ϕ is upward planar with the same external face G_ϕ then we can update F in constant time.*

The deletion of an edge e will either merge two faces f_1 and f_2 in G_ϕ to form a face f in G'_ϕ , or when one of the end vertices of e has a degree of 1 and is incident to a single face f then the facial boundary of f will change. Moreover, we let α_u and α_v represent the angle that is created at u and v respectively as a result of the edge deletion. We say that with the deletion of an edge e from G_ϕ , F will change to the face-sink graph F' . Let T'_f be the tree in F' that contains \bar{f} . T'_f is formed by merging trees in a set $\mathcal{M} \subset F, |\mathcal{M}| \geq 1$ and making some local changes in this merged tree. F' will always satisfy Theorem 1. When all trees in \mathcal{M} have one internal vertex then the resulting tree T'_f will also have exactly one internal vertex. However, if \mathcal{M} contains \hat{T} , the tree in F with no internal vertex, then T'_f will also have no internal vertex. We let μ' denote either the internal vertex in T'_f or the vertex \bar{h} that corresponds to the external tree.

Lemma 6. *Let G_ϕ be an upward planar embedded digraph with a single-source s_G . If we delete an edge $e = (u, v)$ to create an embedded digraph G'_ϕ with a single source $s_{G'}$ then we can update F in constant time.*

Hence we conclude that Algorithm 1 will take $O(\log n)$ -time leading to the following theorem.

Theorem 6. *Let G_ϕ be an upward planar embedded digraph with a single-source s_G . If we add or delete an edge e to create an embedded digraph G'_ϕ with a single source $s_{G'}$ then we can check the upward planarity of G'_ϕ in $O(\log n)$ when the external face is fixed.*

5 Open Problems

As further work, we want to investigate if there is a dynamic upward planarity testing algorithm for embedded digraphs that allows for the external face to change. Moreover, it will be interesting to investigate the optimality of our algorithm. Our algorithm may also be relevant to finding a maximum upward planar subgraph of a single-source embedded digraph and we intend investigating this. A slightly more difficult open problem is to develop a dynamic upward planarity testing algorithm for a single-source digraph over all its embeddings.

References

1. Bender, M.A., Cole, R., Demaine, E.D., Farach-Colton, M., Zito, J.: Two simplified algorithms for maintaining order in a list. In: Möhring, R.H., Raman, R. (eds.) *ESA 2002*. LNCS, vol. 2461, pp. 152–164. Springer, Heidelberg (2002)
2. Bertolazzi, P., Battista, G.D., Didimo, W.: Quasi-upward planarity. *Algorithmica* 32(3), 474–506 (2002)
3. Bertolazzi, P., Battista, G.D., Liotta, G., Mannino, C.: Upward drawings of triconnected digraphs. *Algorithmica* 12(6), 476–497 (1994)
4. Bertolazzi, P., Battista, G.D., Mannino, C., Tamassia, R.: Optimal upward planarity testing of single-source digraphs. *SIAM J. Comput.* 27(1), 132–169 (1998)
5. Demetrescu, C., Finocchi, I., Italiano, G.: Handbook of Graph Theory. In: Yellen, J., Gross, J.L. (eds.) *Dynamic Graph Algorithms*. CRC Press Series, in *Discrete Mathematics and Its Applications*, vol. 10.2 (2003) ISBN 1-58488-090-2
6. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, Englewood Cliffs (1999)
7. Didimo, W.: Computing upward planar drawings using switch-regularity heuristics. In: *SOFSEM*, pp. 117–126 (2005)
8. Didimo, W., Giordano, F., Liotta, G.: Upward spirality and upward planarity testing. In: Healy, P., Nikolov, N.S. (eds.) *GD 2005*. LNCS, vol. 3843, pp. 117–128. Springer, Heidelberg (2006)
9. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.* 31(2), 601–625 (2001)
10. Hutton, M.D., Lubiw, A.: Upward planar drawing of single-source acyclic digraphs. *SIAM J. Comput.* 25(2), 291–311 (1996)
11. Papakostas, A.: Upward planarity testing of outerplanar DAGs. In: *Proceedings Graph Drawing*, pp. 298–306 (1994)
12. Tamassia, R.: On-line planar graph embedding. *J. Algorithms* 21(2), 201–239 (1996)