Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 25, no. 1, pp. 151–170 (2021)
DOI: 10.7155/jgaa.00554

# Multilevel Planarity

*Lukas Barth* *Guido Brückner* *Paul Jungeblut* *Marcel Radermacher*

Karlsruhe Institute of Technology, Germany

Submitted: January 2020 Reviewed: December 2020 Revised: January 2021
Accepted: January 2021 Final: January 2021 Published: January 2021
Article type: Regular paper Communicated by: G. Di Battista

**Abstract.** In this paper, we introduce and study *multilevel planarity*, a generalization of upward planarity and level planarity. Let $G = (V, E)$ be a directed graph and let $\ell : V \to \mathcal{P}(\mathbb{Z})$ be a function that assigns a finite set of integers to each vertex. A multilevel-planar drawing of $G$ is a planar drawing of $G$ such that for each vertex $v \in V$ its $y$-coordinate $y(v)$ is in $\ell(v)$, and each edge is drawn as a strictly $y$-monotone curve.

We present linear-time algorithms for testing multilevel planarity of embedded graphs with a single source and of oriented cycles. Complementing these algorithmic results, we show that multilevel-planarity testing is NP-complete even in very restricted cases.

## 1 Introduction

Testing a given graph for planarity, and, if the graph is planar, finding a planar drawing of it, are classic algorithmic problems. However, one is often not interested in just any planar drawing, but in one that has some additional properties. Examples of such properties include that a given existing partial drawing should be extended [3, 21] or that some parts of the graph should appear clustered together [11, 22]. There also exist notions of planarity specifically tailored to directed graphs. An *upward-planar drawing* is a planar drawing where each edge is drawn as a strictly $y$-monotone curve. While testing upward planarity of a graph is an NP-complete problem in general [16], efficient algorithms are known for outerplanar graphs, single-source graphs and for embedded graphs [27, 6, 20, 7]. One notable restricted variant of upward planarity is that of level planarity. A *level graph* is a directed graph $G = (V, E)$ together with a level assignment $\gamma : V \to \mathbb{Z}$ that assigns an integer level to each vertex and satisfies $\gamma(u) < \gamma(v)$ for all $(u, v) \in E$. A drawing of $G$ is *level planar* if it is upward planar, and for each vertex $v \in V$ the $y$-coordinate of $v$ is $\gamma(v)$. Level-planarity testing and drawing is feasible in linear time [23]. There exist further level-planarity variants on the cylinder and on the torus [1, 4] and there has been considerable research on further-constrained versions of level planarity. Examples include ordering the vertices on each level according to

*E-mail addresses:* lukas.barth@kit.edu (Lukas Barth) brueckner@kit.edu (Guido Brückner) paul.jungeblut@kit.edu (Paul Jungeblut) radermacher@kit.edu (Marcel Radermacher)

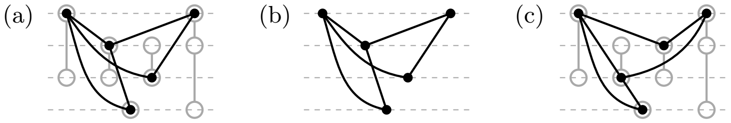


Figure 1: A drawing that is not multilevel-planar (a) since the underlying level graph (b) is not level planar. Changing the $y$-coordinates of two vertices yields a multilevel-planar drawing (c).

so-called constraint trees [2, 17], clustered level planarity [2, 14], partial level planarity [8] and ordered level planarity [25]. Finally, an undirected graph $G = (V, E)$ is *leveled planar* if there exists an assignment $\gamma : V \to \mathbb{Z}$ and a planar drawing $\Gamma$ of $G$ where for each vertex $v \in V$ its $y$-coordinate is $\gamma(v)$ and for each edge $\{u, v\}$ it is $|\gamma(u) - \gamma(v)| = 1$. Recognizing leveled planar graphs is $\mathsf{NP}$-complete [19]. Recently, relationships between track layouts, layered pathwidth and leveled planarity have been studied, and bipartite outerplanar graphs and squaregraphs have been shown to be leveled planar [5].

**Contribution and Outline.** In this paper, we introduce and study *multilevel planarity.* Let $\mathcal{P}(\mathbb{Z})$ denote the power set of integers. The input of the multilevel-planarity testing problem consists of a directed graph $G = (V, E)$ together with a function $\ell : V \to \mathcal{P}(\mathbb{Z})$, called a multilevel assignment, which assigns admissible levels, represented as a set of integers, to each vertex. A multilevel-planar drawing of $G$ is a planar drawing of $G$ such that for the $y$-coordinate of each vertex $v \in V$ it holds that $y(v) \in \ell(v)$, and each edge is drawn as a strictly $y$-monotone curve. See Figure 1 for an example of a multilevel-planar graph. For each vertex $v$ the gray circles in the same column as $v$ visualize the set $\ell(v)$. The choice of $y$-coordinates shown in (a) is not level planar (b), but the choice of $y$-coordinates shown in (c) gives a multilevel-planar drawing. Figure 2 showcases one of the use cases of multilevel planarity. It visualizes an excerpt of the genealogy of European royalty [28, 13, 24]. In this graph, vertices are associated with individuals. The intervals assigned to a vertex are derived from the corresponding individual's lifespan and visualized as gray bars. Dashed edges represent marriages and solid lines connected to dashed edges represent descent of children. To improve readability, the edges representing marriages are drawn as horizontal line segments even though the formal definition requires *strictly* $y$-monotone curves.

Our paper is structured as follows. We start by discussing some preliminaries, including the relationship between multilevel planarity and existing planarity variants in Section 2. Then, we present linear-time algorithms that test multilevel planarity of embedded single-source graphs and of oriented cycles with multiple sources in Sections 3 and 4, respectively. In Section 5, we complement these algorithmic results by showing that multilevel-planarity testing is $\mathsf{NP}$-complete for abstract single-source graphs, for oriented trees and for embedded multi-source graphs where $|\ell(v)| \leq 2$ for all $v \in V$. We finish with some concluding remarks in Section 6.

## 2 Preliminaries

This section consists of three parts. First, we introduce basic terminology and notation. Second, we discuss the relationship between multilevel planarity and existing planarity variants for directed graphs. Third, we define a normal form for multilevel assignments, which simplifies the arguments in Sections 3 and 4.

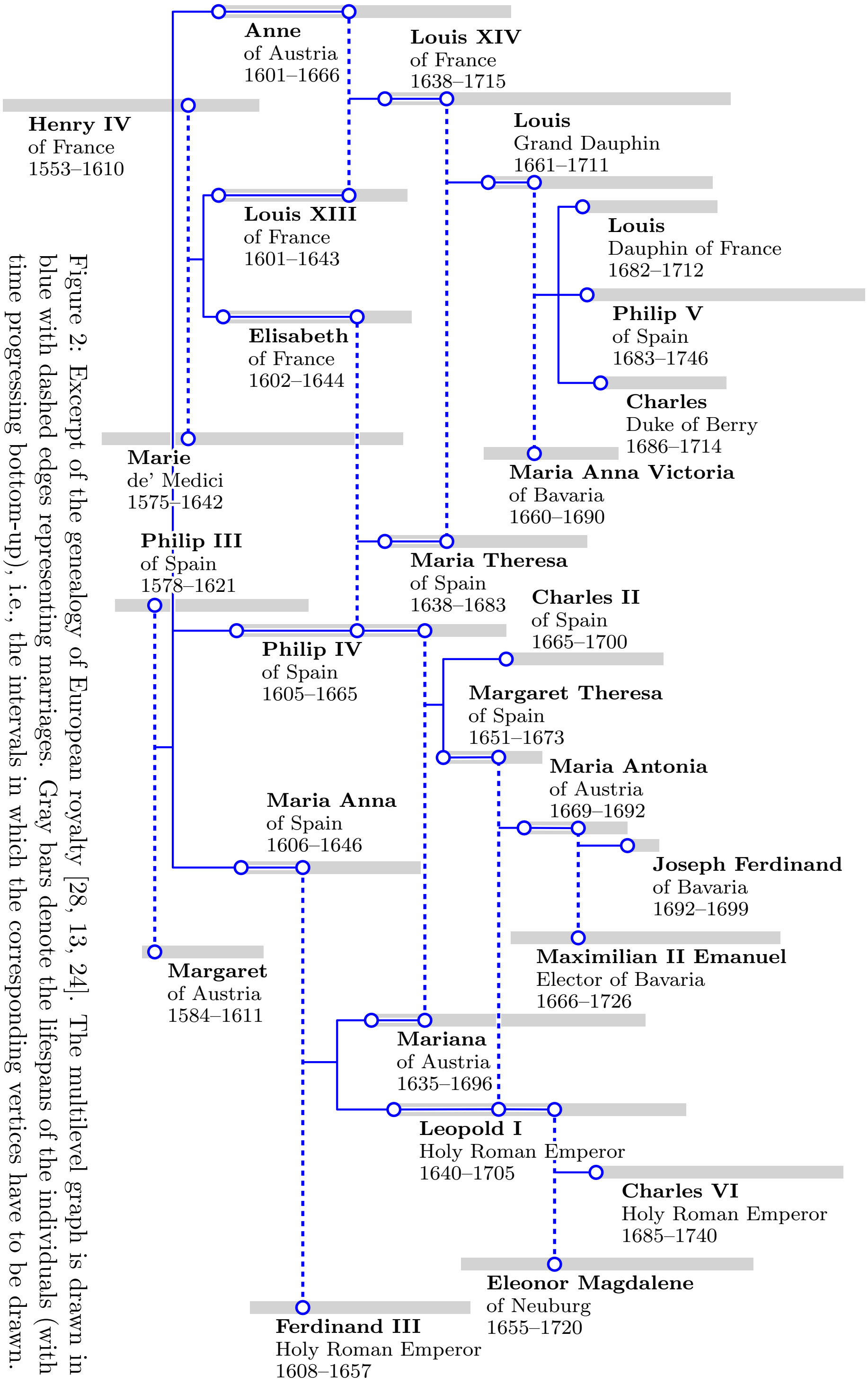


Figure 2: Excerpt of the genealogy of European royalty [28, 13, 24]. The multilevel graph is drawn in blue with dashed edges representing marriages. Gray bars denote the lifespans of the individuals (with time progressing bottom-up), i.e., the intervals in which the corresponding vertices have to be drawn.

**Basic Terminology.** Let $G = (V, E)$ be a directed graph. A *drawing* of $G$ maps each vertex in $V$ to a point in the Euclidean plane and each edge $e \in E$ to a Jordan curve between the two endpoints of $e$. A drawing is *planar* if Jordan curves corresponding to distinct edges do not intersect, except, possibly, in common endpoints. A planar drawing defines a cyclic order of the incident edges around each vertex. A *combinatorial embedding* assigns such a cyclic order to each vertex of a graph. We say that two planar drawings are *combinatorially equivalent* if they define the same combinatorial embedding and have the same outer face. A drawing of $G$ is *upward* if each edge is drawn as a strictly $y$-monotone curve. A *multilevel assignment* $\ell : V \to \mathcal{P}(\mathbb{Z})$ assigns a finite set of integers to each vertex. An upward-planar drawing is *multilevel planar* if $y(v) \in \ell(v)$ for all $v \in V$. Note that any finite set of integers can be represented as a finite list of finite integer intervals. We choose this representation to be able to represent sets of integers that contain large intervals of numbers more efficiently.

A vertex of a directed graph with no incoming (outgoing) edges is a *source* (*sink*). A directed, acyclic and planar graph with a single source $s$ is an *sT-graph.* An $sT$-graph with a single sink $t$ and an edge $(s, t)$ is an *st-graph.* In any upward-planar drawing of an $st$-graph, the unique source and sink are the lowest and highest vertices, respectively, and both are incident to the outer face. For a face $f$ of a planar drawing, an incident vertex $v$ is called *source switch* (*sink switch*) if all edges incident to $f$ and $v$ are outgoing (incoming). Note that a source switch or a sink switch does not need to be a source or a sink in $G$. We will frequently add incoming edges to sources and outgoing edges to sinks during later constructions, referring to this as *source canceling* and *sink canceling*, respectively. An *oriented path* of length $k$ is a sequence of vertices $(v_1, v_2, \dots, v_{k+1})$ such that for all $1 \leq i \leq k$ either the edge $(v_i, v_{i+1})$ or the edge $(v_{i+1}, v_i)$ is in $G$. A *directed path* of length $k$ is a sequence of vertices $(v_1, v_2, \dots, v_{k+1})$ such that for all $1 \leq i \leq k$ the edge $(v_i, v_{i+1})$ is in $G$. Let $u, v \in V$ be two distinct vertices. Vertex $u$ is a *descendant* of $v$ in $G$, if there exists a directed path from $v$ to $u$. A *topological ordering* is a function $\tau : V \to \mathbb{N}$ such that for every $v \in V$ and for each descendant $u$ of $v$ it is $\tau(v) < \tau(u)$.

**Relationship to Existing Planarity Variants.** Multilevel-planarity testing is a generalization of level planarity testing. To see this, let $G = (V, E)$ be a directed graph together with a level assignment $\gamma : V \to \mathbb{Z}$. Define $\ell(v) = \{\gamma(v)\}$ for all $v \in V$. It is readily observed that a drawing $\Gamma$ of $G$ is level planar with respect to $\gamma$ if and only if $\Gamma$ is multilevel planar with respect to $\ell$. Therefore, level planarity reduces to multilevel planarity in linear time.

Multilevel-planarity testing is also a generalization of upward planarity testing. Without loss of generality, the vertices in an upward-planar drawing can be assigned unique integer $y$-coordinates so that there is at least one vertex on each level in $[1, |V|]$. Hence, upward planarity of $G$ can be tested by setting $\ell(v) = [1, |V|]$ for all $v \in V$ and testing the multilevel planarity of $G$ with respect to $\ell$. Therefore, upward planarity reduces to multilevel planarity in linear time. By then restricting the multilevel assignment, multilevel planarity can also be seen as a constrained version of upward planarity. Garg and Tamassia [16] showed that upward-planarity testing is NP-complete. It is easy to see that multilevel-planarity testing is in NP and so we conclude the following.

**Theorem 1** *Multilevel-planarity testing is NP-complete.*

Multilevel planarity is related to leveled planarity. Both notions ask about the existence of a certain $y$-coordinate assignment $\gamma$. However, multilevel planarity is defined for directed graphs, i.e., for two adjacent vertices it is known which one has the greater $y$-coordinate, whereas leveled planarity is defined for undirected graphs. And for adjacent vertices $u, v$ in a leveled planar drawing it must be $|\gamma(u) - \gamma(v)| = 1$, whereas no such restriction exists for the multilevel-planar drawings.

**Multilevel Assignment Normal Form.** A multilevel assignment $\ell$ has *normal form* if for all $(u, v) \in E$ it is $\min \ell(u) < \min \ell(v)$ and $\max \ell(u) < \max \ell(v)$. Some proofs are easier to follow for multilevel assignments in normal form. The following lemma justifies that we may assume without loss of generality that $\ell$ has normal form.

**Lemma 1** *Let $G = (V, E)$ be a directed graph together with a multilevel assignment $\ell$. Then there exists a multilevel assignment $\ell'$ in normal form such that a drawing of $G$ is multilevel planar with respect to $\ell$ if and only if it is multilevel planar with respect to $\ell'$. Moreover, $\ell'$ can be computed in linear time.*

**Proof:** The idea is to convert $\ell(v)$ into $\ell'(v) \subseteq \ell(v)$ for all $v \in V$ by finding a lower bound $l_v$ and an upper bound $u_v$ for the level of $v$, and then setting $\ell'(v) = \ell(v) \cap [l_v, u_v]$. If this set is empty there exists no multilevel-planar drawing. To find the lower bound, iterate over the vertices in increasing order with respect to some topological ordering $\tau$ of $G$. Because all edges have to be drawn as strictly $y$-monotone curves, for each vertex $v \in V$ it must be $y(v) > \max_{(w,v)\in E} l_w$. So, set $l_v = \max(\min \ell(v), \max_{(w,v)\in E} l_w + 1)$. Analogously, to find the upper bound, iterate over $V$ in decreasing order with respect to $\tau$. Again, because edges are drawn as strictly $y$-monotone curves, for each vertex $v \in V$ it must hold true that $y(v) < \min_{(v,w)\in E} u_w$. Therefore, set $u_v = \min(\max \ell(v), \min_{(v,w)\in E} u_w - 1)$. This means that in any multilevel-planar drawing of $G$ the $y$-coordinate of $v \in V$ is $y(v) \in \ell(v) \cap [l_v, u_v]$. So it follows that a drawing of $G$ is multilevel planar with respect to $\ell$ if and only if it is multilevel planar with respect to $\ell'$.

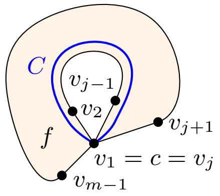
To see that the running time is linear, note that a topological ordering of $G$ can be computed in linear time and every vertex and edge is handled at most twice during the procedure described above. Because every level candidate in $\ell$ is removed at most once, the total running time is $\mathcal{O}(n + \sum_{v\in V} |\ell(v)|)$, i.e., linear in the size of the input. □

## 3 Embedded *sT*-Graphs

In this section, we characterize multilevel-planar $sT$-graphs as subgraphs of certain planar $st$-graphs. Similar characterizations exist for upward planarity and level planarity [12, 26]. The idea behind our characterization is that for any given multilevel-planar drawing, we can find a set of edges that can be inserted without rendering the drawing invalid, and which make the underlying graph an $st$-graph. For these $st$-graphs multilevel planarity and planarity coincide and we can use existing linear-time algorithms to find (multilevel-)planar drawings. This technique is similar to the one found by Bertolazzi et al. [7], and in fact is built on top of it.

To use this characterization for multilevel-planarity testing, we do not require that a multilevel-planar drawing is given. We show that if we choose the set of edges to be inserted carefully, the respective set of edges can be inserted into *any* multilevel-planar drawing for a fixed combinatorial embedding. An algorithm constructing such an edge set can therefore be used to test for multilevel planarity of embedded $sT$-graphs, resulting in Theorem 2. The algorithm is constructive in the sense that it finds a multilevel-planar drawing if one exists. In Section 5, we show that testing multilevel planarity of $sT$-graphs without a fixed combinatorial embedding is NP-hard. Recall that every multilevel-planar drawing is upward planar. We now prove that the vertex with the largest $y$-coordinate on the boundary of each face is the same across all combinatorially equivalent drawings. To this end we use the notion of large and small angles at sink switches that Bertolazzi et al. [7] have used for biconnected graphs. We extend this notion to simply-connected $sT$-graphs. Lemma 2 is used to argue that such angles are well-defined, Lemma 3 extends the observations of Bertolazzi et al. to simply-connected graphs and Lemma 4 sets the foundation to our $st$-graph extension.

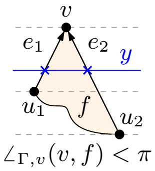
**Lemma 2** *Let $G = (V, E)$ be an $sT$-graph together with an upward-planar drawing $\Gamma$. Further, let $f$ be a face of $\Gamma$ and let $c$ denote a sink switch of $f$. Then $c$ appears exactly once on the cyclic walk around the boundary of $f$.*

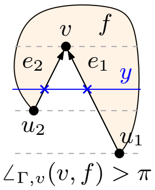


**Proof:** Let $w = [v_1 = c, v_2, \ldots, v_j = c, \ldots, v_m = v_1 = c]$ be the cyclic walk around $f$, where $2 < j < m - 1$. We show that $c$ is not a sink switch. Construct a Jordan curve $C$ by closely following $w$ along $v_1 = c, v_2, \ldots, v_j = c$ within $f$ and then crossing through $c$ to close the curve. Let $G_{\text{outer}}$ denote the subgraph of $G$ induced by the vertices in the exterior of $C$ and on $C$. Let $G_{\text{inner}}$ denote the subgraph of $G$ induced by the vertices in the interior of $C$ and on $C$.

Consider the case that $H \in \{G_{\text{outer}}, G_{\text{inner}}\}$ does not contain $s$. Because $G$ is an $sT$-graph there exists in $G$ a directed path from $s$ to every vertex in $H$. Because no edge crosses $C$ and the only vertex that lies on $C$ is $c$ all these paths contain $c$. Therefore, there exists in $H$ a directed path from $c$ to every vertex of $H$, i.e., $H$ is an $sT$-graph with source $c$. Either $v_2$ or $v_{j+1}$ lies in $H$ by construction and so one of the edges $(v_1 = c, v_2), (v_j = c, v_{j+1})$ exists in $G$, i.e., $c$ is not a sink switch of $f$.

Now consider the other case, namely that both $G_{\text{outer}}$ and $G_{\text{inner}}$ contain $s$. By construction $G_{\text{outer}}$ and $G_{\text{inner}}$ share only the vertex $c$, i.e., $c = s$. Because $s$ is the source of $G$ it cannot be a sink switch of any face. □





Let $v$ be a sink switch of a face $f$ in $\Gamma$. Further, let $e_1 = (u_1, v), e_2 = (u_2, v)$ denote the edges incident to $f$ and $v$. By Lemma 2 the choice of these edges is unique. Let $f$ lie to the right of $e_1$ and to the left of $e_2$ with respect to the directions of those edges. Because $\Gamma$ is upward there exists a horizontal line $y$ that intersects both $e_1$ and $e_2$ exactly once but does not intersect $v$. For $i = 1, 2$, let $x_i$ denote the $x$-coordinate of the intersection of $y$ and $e_i$. Define $\angle_{\Gamma,f}(v)$ as *small* (written as $\angle_{\Gamma,f}(v) < \pi$) if $x_1 < x_2$ (see the upper part of the figure on the right) and as *large* (written as $\angle_{\Gamma,f}(v) > \pi$) if $x_1 \geq x_2$ (see the lower part of the figure on the right). Note that $x_1 = x_2$ implies $e_1 = e_2$, i.e., $u_1 = u_2$ is a cutvertex. This sets the stage for the following.

**Lemma 3** *Let $G$ be an $sT$-graph together with an upward-planar drawing $\Gamma$. Then the following properties hold:*

1. *For each sink switch $v$ on the boundary of the outer face $h$ it is $\angle_{\Gamma,h}(v) > \pi$.*

2. *For each inner face $f$ of $\Gamma$ there is exactly one sink switch $t_f$ on the boundary of $f$ with $\angle_{\Gamma,f}(t_f) < \pi$, namely the vertex with greatest $y$-coordinate among all vertices incident to $f$.*

**Proof:** Use induction over the number of biconnected components of $G$. In the base case of a biconnected graph both properties were observed by Bertolazzi et al. [7, page 138, Facts 2 and 3].

Let $c$ be a cutvertex of $G$. Then there exists a face $f$ of $\Gamma$ such that $c$ appears more than once on the cyclic walk around $f$. Let $w = [v_1 = c, v_2, \ldots, v_j = c, \ldots, v_m = v_1 = c]$ denote the cyclic walk around $f$, where $2 < j < m - 1$. Further, let $C, G_{\text{outer}}, G_{\text{inner}}$ be defined as in the proof of Lemma 2.

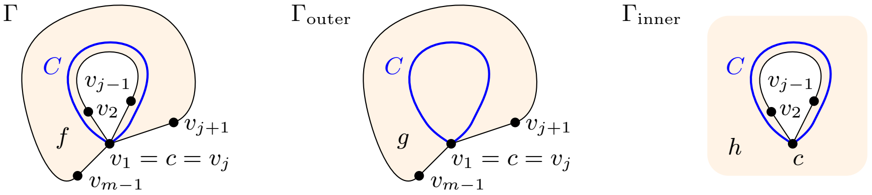


Figure 3: Proof of Lemma 3. A drawing $\Gamma$ of $G$ can be separated into drawings $\Gamma_{\text{outer}}, \Gamma_{\text{inner}}$ of two $sT$-graphs with fewer biconnected components.

First, we show that $G_{\text{outer}}, G_{\text{inner}}$ are $sT$-graphs. If $H \in \{G_{\text{outer}}, G_{\text{inner}}\}$ does not contain $s$, we have shown in the proof of Lemma 2 that $H$ is an $sT$-graph with source $c$. Consider the case that $H$ contains $s$. Let $H' \in \{G_{\text{outer}}, G_{\text{inner}}\}$ with $H' \neq H$. Because $G$ is an $sT$-graph there exists in $G$ a directed path from $s$ to every vertex in $H$. Suppose that such a directed path $p$ contains a vertex not in $H$, i.e., a vertex in $H'$ other than $c$. Because no edge crosses $C$ and the only vertex that lies on $C$ is $c$ this means that $p$ must contain $c$ twice. This contradicts the fact that $G$ is acyclic. Hence, there exists in $H$ a directed path from $s$ to every vertex of of $H$, i.e., $H$ is an $sT$-graph with source $s$.

The drawing $\Gamma$ induces drawings $\Gamma_{\text{outer}}, \Gamma_{\text{inner}}$ of $G_{\text{outer}}, G_{\text{inner}}$, respectively. We have just shown that $G_{\text{outer}}, G_{\text{inner}}$ are $sT$-graphs. Each has strictly fewer biconnected components than $G$, so Properties 1 and 2 hold by induction.

Observe that $\Gamma$ can be obtained from $\Gamma_{\text{outer}}$ by inserting $\Gamma_{\text{inner}}$ into $C$; see Figure 3. This changes no angles, except at $c$ which is not a sink switch of $f$ by Lemma 2. Every face of $\Gamma$ except for $f$ exists in one of $\Gamma_{\text{outer}}, \Gamma_{\text{inner}}$ and the claimed properties are true by induction. Let $g$ denote the face of $\Gamma_{\text{outer}}$ that contains the interior of $C$. Let $h$ denote the outer face of $\Gamma_{\text{inner}}$, i.e., the face of $\Gamma_{\text{inner}}$ that contains the exterior of $C$. Face $f$ in $\Gamma$ is obtained from combining the faces $g$ in $\Gamma_{\text{outer}}$ and $h$ of $\Gamma_{\text{inner}}$. Every sink switch $v$ on the boundary of $f$ in $\Gamma$ is a sink switch on the boundary of either $g$ in $\Gamma_{\text{outer}}$ or $h$ in $\Gamma_{\text{inner}}$. If $v$ is a sink switch on the boundary of $h$ Property 1 implies $\angle_{\Gamma_{\text{inner}},h}(v) > \pi$ and therefore $\angle_{\Gamma,f}(v) > \pi$. Now consider the case that $v$ is a sink switch on the boundary of $g$. If $g$ is the outer face of $\Gamma_{\text{outer}}$ then $f$ is the outer face of $\Gamma$. Then Property 1 implies $\angle_{\Gamma_{\text{outer}},g}(v) > \pi$ and therefore $\angle_{\Gamma,f}(v) > \pi$. If $g$ is an inner face of $\Gamma_{\text{outer}}$ then $f$ is an inner face of $\Gamma$. Property 2 implies that there is exactly one sink switch $v$ on the boundary of $g$ in $\Gamma_{\text{outer}}$ with $\angle_{\Gamma_{\text{outer}},g}(v) < \pi$, namely the vertex with the greatest $y$-coordinate among all vertices incident to $g$. Then $v$ is the only sink switch on the boundary of $f$ in $\Gamma$ with $\angle_{\Gamma,f}(v) < \pi$. Also, because $C$ is contained within $g$ vertex $v$ is the vertex with the greatest $y$-coordinate among all vertices incident to $f$. □

We are now ready to prove the following.

**Lemma 4** *Let $G$ be an $sT$-graph, let $\Gamma, \Gamma'$ be combinatorially equivalent upward-planar drawings of $G$ and let $f$ be an inner face of $\Gamma$ and $\Gamma'$. Then the vertex with the greatest $y$-coordinate among all vertices incident to $f$ is the same in $\Gamma$ and $\Gamma'$.*

**Proof:** Let $t_f$ denote the vertex with the greatest $y$-coordinate among all vertices incident to $f$ in $\Gamma$. Further let $e_1 = (v_1, t_f)$ and $e_2 = (v_2, t_f)$ be the edges incident to $f$ and $t_f$ (by Lemma 2 and because $t_f$ is a sink switch $e_1, e_2$ are the only such edges). Property 2 of Lemma 3 states that $\angle_{\Gamma,f}(v) < \pi$, i.e., $e_1 \neq e_2$. Assume that $t_f$ does not have the greatest $y$-coordinate of all vertices incident to $f$ in $\Gamma'$. See Figure 4. Because $G$ has a single source $s$, there exist directed

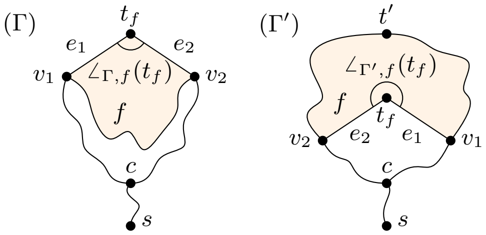


Figure 4: Proof of Lemma 4.

paths $p_1$ and $p_2$ from $s$ to $v_1$ and $v_2$, respectively. Then the left-to-right order of the edges $e_1$ and $e_2$ in $\Gamma$ and $\Gamma'$ is determined by the order of the outgoing edges at the last common vertex $c$ on $p_1$ and $p_2$. Because $e_1 \neq e_2$ it is $c \neq t_f$. Let $t' \neq t_f$ be the vertex with greatest $y$-coordinate of all vertices incident to $f$ in $\Gamma'$. Then it holds that $\angle_{\Gamma',f}(t') < \pi$ and from Property 2 of Lemma 3 it follows that $\angle_{\Gamma',f}(t_f) > \pi$. Since $\Gamma$ and $\Gamma'$ have the same underlying combinatorial embedding, the clockwise cyclic walk around $f$ is identical in both drawings. But because $\angle_{\Gamma,f}(t_f) < \pi$ and $\angle_{\Gamma',f}(t_f) > \pi$, the left-to-right order of the outgoing edges of $c$ is different in $\Gamma$ and $\Gamma'$. This means that $\Gamma$ and $\Gamma'$ are not combinatorially equivalent: If $c$ has an incoming edge the cyclic order of the edges around $c$ is different in $\Gamma$ and $\Gamma'$. Otherwise it is $s = c$ and because $(s,t)$ is the left-most edge by definition the cyclic order of the edges around $c$ is different in $\Gamma$ and $\Gamma'$. □

Bertolazzi et al. showed that any $sT$-graph with an upward-planar drawing can be extended to an $st$-graph with an upward-planar drawing that extends the original drawing [6, 7]. More formally, let $G = (V, E)$ be an $sT$-graph together with an upward-planar drawing $\Gamma$. Then there exists an $st$-graph $G_{st} = (V \dot\cup \{t\}, E \dot\cup E_{st})$ where $t$ is the unique sink together with an upward-planar drawing $\Gamma_{st}$ that extends $\Gamma$. Moreover, $G_{st}$ and $\Gamma_{st}$ can be computed in linear time. Note that in general it is possible for a given $E_{st}$ to choose an upward-planar drawing $\Gamma$ of $G$ so that the additional edges in $E_{st}$ cannot be added into $\Gamma$ as $y$-monotone curves. For an example, see Figure 5, where augmenting with the red and black edge works for the drawing shown in (a) but not for the one shown in (b), whereas augmenting with the blue and black edge works for both drawings. In Lemma 5 we show that there is a set $E_{st}$ that can be added into any upward planar drawing with the same combinatorial embedding as $\Gamma$. In a way, this is the most general set $E_{st}$.

**Lemma 5** *Let $G = (V, E)$ be a directed $sT$-graph with a fixed combinatorial embedding. Then there exists an $st$-graph $G_{st} = (V \dot\cup \{t\}, E \dot\cup E_{st})$, where $t$ is the unique sink, such that for any upward-planar drawing $\Gamma$ of $G$ there exists an upward-planar drawing $\Gamma_{st}$ of $G_{st}$ that extends $\Gamma$. Moreover, $G_{st}$ can be computed in linear time.*

**Proof:** Start by finding an initial upward-planar drawing $\Gamma$ of $G$ in linear time using the algorithm due to Bertolazzi et al. [7]. The algorithm additionally outputs an embedded $st$-graph $G^0_{st} = (V \dot\cup \{t\}, E \dot\cup E^0_{st})$ together with an upward-planar drawing $\Gamma^0_{st}$ that extends $\Gamma$. This means that $\Gamma^0_{st}$ is obtained from $\Gamma$ by drawing each edge $(u,v) \in E^0_{st}$ within some face $F^0(u,v)$ of $\Gamma$. Define a function $T$ that maps the faces of $\Gamma$ to the vertex set $V \dot\cup \{t\}$ as follows. If $f$ is the outer face of $\Gamma$ define $T(f) = t$ and if $f$ is an inner face of $\Gamma$ define $T(f) = t_f$, where $t_f$ denotes the sink switch of $f$ with $\angle_{\Gamma,f}(t_f) < \pi$, which is unique by Property 2 of Lemma 3 and the same for all combinatorially equivalent drawings by Lemma 4. We show that $E_{st} = \{(u, T(F^0(u,v))) \mid (u,v) \in E^0_{st}\}$ satisfies the claim.

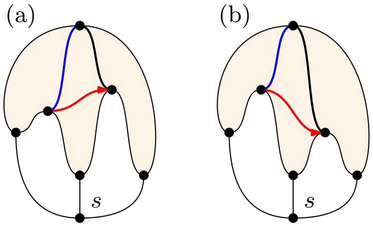


Figure 5: Some edges are not admissible for augmentation in Lemma 5.

For each face $f$ of $\Gamma$ the vertex $T(f)$ has the greatest $y$-coordinate among all vertices incident to $f$. This means that for each edge $(u, v) \in E_{st}$ the $y$-coordinate of $u$ is smaller than the $y$-coordinate of $v$ in $\Gamma$. Together with the fact that $\Gamma$ is an upward-planar drawing this means that $G_{st}$ is acyclic. Recall that $G^0_{st}$ is an $st$-graph, i.e., every sink of $G$ is incident to an outgoing edge in $E^0_{st}$. By construction of $E_{st}$ every sink of $G$ is then also incident to an outgoing edge in $E_{st}$. Therefore $G_{st}$ has exactly one sink, namely $t$. Together with the fact that $G_{st}$ is acyclic this shows that $G_{st}$ is an $st$-graph.

Now let $\Gamma'$ be an upward-planar drawing of $G$. Recall that the embedding of $G$ is fixed, so $\Gamma'$ is combinatorially equivalent to $\Gamma$. We show that we can extend $\Gamma'$ to an upward-planar drawing of $G_{st}$. To this end, we insert the vertex $t$ and then insert the edges in $E_{st}$ into $\Gamma'$ one after the other. Let $Y$ denote the greatest $y$-coordinate of any vertex in $\Gamma'$. Insert $t$ into $\Gamma'$ with $y$-coordinate $Y + 1$.

Let $(u, v)$ be an edge in $E_{st}$. The idea is that it is possible to walk from any vertex $x$ on the boundary of $f$ upwards to $t_f$. If $x$ is incident to an outgoing edge $(x, w)$ incident to $f$ follow that edge to $w$. Because $\Gamma'$ is an upward-planar drawing this segment is $y$-monotone. Then continue walking up from $w$ to $t_f$. Otherwise, if $x$ is not incident to an outgoing edge incident to $f$, it is a sink switch of $f$. There are two cases.

1. It is $\angle_{\Gamma',f}(x) > \pi$. Then walk up vertically from $x$ within $f$. If $f$ is an inner face of $\Gamma'$, this walk will meet either a vertex $w$ or an edge $(x', w)$. In the former case continue walking up from $w$. The latter case has two subcases. The first subcase is $(x', w) \notin E_{st}$. Then $(x', w)$ is incident to $f$, so follow the edge to $w$ and then continue walking up from $w$. The second subcase is $(x', w) \in E_{st}$. Then $(x', w)$ is an edge that was inserted into $f$ in $\Gamma'$ previously. Note that all edges inserted into $f$ have endpoint $t_f$, i.e., $w = t_f = v$, so follow $(x', w)$ to its endpoint $w$ to complete the drawing of $(u, v)$.

   If $f$ is the outer face of $\Gamma'$, either one of the previous situations occurs, or we could walk vertically up infinitely without meeting an edge or a vertex. Note that this means $v = t$. In this case stop walking up when the $y$-coordinate is $Y$, bend and then connect to $t$ with a straight line segment.

2. It is $\angle_{\Gamma',f}(x) < \pi$. From Lemma 3 it follows that $x = t_f$.

□

We now have a set of edges that can be used to complete $G$ into $G_{st}$. If a multilevel-planar drawing for the given combinatorial embedding of $G$ respecting $\ell$ exists, then it must also exist for $G_{st}$. However, the property of $\ell$ being in normal form might not be fulfilled anymore in $G_{st}$ because of the added edges. We therefore need to bring $\ell$ into normal form $\ell'$ again. Lemma 1 tells us that this does not impact multilevel planarity. We conclude that $G$ is multilevel planar with respect to $\ell$ if and only if $G_{st}$ is multilevel planar with respect to $\ell'$. The final property we need is that $G_{st}$ is level planar with respect to any level assignment $\gamma$ (recall that $\gamma$ is a level assignment of $G_{st}$ if for each directed edge $(u, v)$ in $G_{st}$ it is $\gamma(u) < \gamma(v)$). The following lemma is due to Leipert [26, Theorem 5.1 and page 121], who notes that an algorithm for drawing upward planar graphs by Di Battista and Tamassia [12, Theorem 3.5] can be adapted for the level-planar setting.

**Lemma 6** *Let $G$ be an $st$-graph on $n$ vertices together with a level assignment $\gamma$. Then for any combinatorial embedding of $G$ there exists a combinatorially equivalent drawing of $G$ that is level planar with respect to $\gamma$. Moreover, such a drawing has $\mathcal{O}(n)$ size and can be computed in $\mathcal{O}(n)$ time.*

If $\ell'$ is in normal form, $\ell'(v) \neq \emptyset$ is a necessary and sufficient condition that there exists a level assignment $\gamma : V \to \mathbb{Z}$ with $\gamma(v) \in \ell'(v)$ for all $v \in V$. Setting $\gamma(v) = \min \ell'(v)$ is one possible such level assignment. Then $G$ is level planar with respect to $\gamma$ and therefore multilevel planar with respect to $\ell$, resulting in the characterization of multilevel-planar $st$-graphs:

**Corollary 1** *Let $G$ be an $st$-graph together with a multilevel assignment $\ell$ in normal form. Then there exists a multilevel-planar drawing for any combinatorial embedding of $G$ if and only if $\ell(v) \neq \emptyset$ for all $v$.*

For a constructive multilevel-planarity testing algorithm, we now first take the edge set computed by the algorithm by Bertolazzi et al. [7] and modify it using Lemma 5 to complete any $sT$-graph to an $st$-graph. Note that for this step, we need a fixed combinatorial embedding to be given, as is required by the second property of Lemma 3. Once arrived at an $st$-graph, we check the premise of Corollary 1. Then, we either output that the graph is not multilevel planar or use Lemma 6 to find a multilevel-planar drawing in linear time. This concludes the testing algorithm:

**Theorem 2** *Let $G$ be an $sT$-graph with a multilevel assignment $\ell$ together with a combinatorial embedding and an outer face. Then it can be decided in linear time whether there exists a combinatorially equivalent multilevel-planar drawing of $G$. If so, such a drawing can be computed within the same running time.*

Our algorithm uses the fact that to augment $sT$-graphs to $st$-graphs, only edges connecting sinks to other vertices need to be inserted. For graphs with multiple sources and multiple sinks, further edges connecting sources to other vertices need to be inserted. The interactions that occur then are very complex: In Section 5, we show that deciding multilevel planarity is NP-complete for embedded multi-source graphs. In the next section, we identify oriented cycles as a class of multi-source graphs for which multilevel planarity can be decided efficiently.

# 4 Oriented Cycles

In this section, we present a constructive multilevel-planarity testing algorithm for oriented cycles, i.e., directed graphs whose underlying undirected graph is a simple cycle. We start by giving a condition for when an oriented cycle $G = (V, E)$ together with some level assignment $\gamma$ admits a level-planar drawing. This condition yields an algorithm for the multilevel-planar setting. In this section, $\gamma$ is always a level assignment and $\ell$ is always a multilevel assignment. Define $\max \gamma = \max\{\gamma(v) \mid v \in V\}$ and $\min \gamma = \min\{\gamma(v) \mid v \in V\}$. Further set $\max \ell = \max\{\max \ell(v) \mid v \in V\}$ and $\min \ell = \min\{\min \ell(v) \mid v \in V\}$. Let $S_{\min} \subset V$ be sources

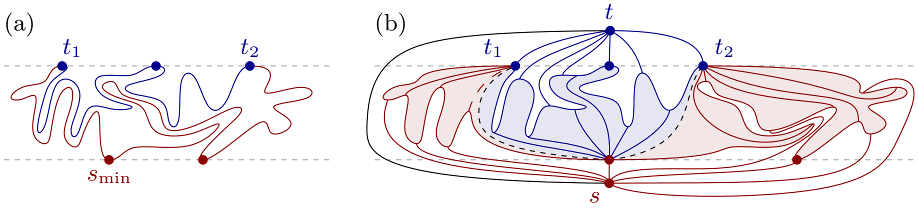


Figure 6: Augmenting the oriented cycle in (a) to an *st*-graph in (b). See the proof of Lemma 7.

with minimal level, i.e., $S_{\min} = \{v \in V \mid \gamma(v) = \min \gamma\}$, and let $T_{\max} \subset V$ be the sinks with maximal level. We call sources in $S_{\min}$ *minimal sources* (these are the sources that must lie on the minimal level), sinks in $T_{\max}$ are *maximal sinks* (these are the sinks that must lie on the maximal level). We say that $\gamma$ is *separating* when there exist two edge-disjoint oriented paths in $G$ so that one contains all minimal sources and the other contains all maximal sinks.

**Lemma 7** *Let $G$ be an oriented cycle with a level assignment $\gamma$. Then $G$ is level planar with respect to $\gamma$ if and only if $\gamma$ is separating.*

**Proof:** The "only if" part is due to Healy et al. [18, Theorem 7]. For the "if" part, augment $G$ to a planar *st*-graph as follows. See Figure 6 (a) for a cycle and (b) for the augmented *st*-graph. Let $p_t$ be the oriented path of minimal length that contains all sinks in $T_{\max}$ and no vertex in $S_{\min}$, and let $t_1, t_2 \in T_{\max}$ denote its endpoints. Fix some vertex $s_{\min} \in S_{\min}$ and cancel every source $v$ on $p_t$ by adding an edge from $s_{\min}$ to $v$. Because $s_{\min} \in S_{\min}$ it is $\gamma(s_{\min}) < \gamma(v)$ and the graph remains acyclic. Introduce a new sink $t$ with $\gamma(t) = \gamma(t_1) + 1$ and cancel every sink $v$ on $p_t$ by adding an edge from $v$ to $t$. Because $\gamma(v) < \gamma(t)$ the graph remains acyclic. Let $p_1$ denote an oriented path from $t_1$ to $s_{\min}$ and let $p_2$ denote an oriented path from $s_{\min}$ to $t_2$ so that $p_1$ and $p_2$ share no edge. Note that the paths $p_t, p_1, p_2$ are pairwise disjoint except in common endpoints. Cancel every sink $v$ on $p_1$ or $p_2$ by adding an edge from $v$ to $t_1$ or $t_2$, respectively. Introduce a new source $s$ with $\gamma(s) = \gamma(s_{\min}) - 1$ and cancel every source $v$ on $p_1$ or $p_2$ by adding an edge from $s$ to $v$. Because $\gamma(s) < \gamma(v)$ the graph remains acyclic. Finally add the edge $(s, t)$ to make the graph an *st*-graph.

To see that the augmented graph is planar observe that the cycle is trivially planar. Furthermore, all augmentation edges incident to $t$ are incident to vertices on $p_t$ and no augmentation edges incident to $s$ are incident to vertices on $p_t$ (except, possibly, for $t_1$ and $t_2$). Moreover, the interior of the circle is partitioned into three regions corresponding to the oriented paths $p_t, p_1, p_2$. In Figure 6 these regions are separated by the black dashed edges, the area corresponding to $p_t$ is shaded in blue whereas the areas corresponding to $p_1$ and $p_2$ are shaded in red. The regions are disjoint and all augmentation edges in one region have the same endpoint, therefore they do not cross. Because $p_t, p_1, p_2$ are disjoint except in common endpoints the augmentation edges of different areas do not cross. Hence, the augmented graph is a planar *st*-graph and then Lemma 6 yields that $G$ is level planar with respect to $\gamma$. □

Recall that any multilevel-planar drawing is a level-planar drawing with respect to some level assignment $\gamma$. Lemma 7 states a necessary and sufficient condition for $\gamma$ so that the drawing is level planar. Given a multilevel assignment $\ell$, we therefore find a separating level assignment $\gamma$, or determine that no such level assignment exists. It must be $\ell(v) \neq \emptyset$ for all $v \in V$; otherwise, $G$ admits no multilevel drawing. We find a level assignment $\gamma$ that keeps the sets $S_{\min}$ and $T_{\max}$ as small as possible, because such a level assignment is, intuitively, most likely to be separating. To this end, let $S_{\mathrm{may}} \subset V$ contain each source $s'$ of $G$ with $\min \ell(s') = \min \ell$. Further, let $S_{\mathrm{must}} \subseteq S_{\mathrm{may}}$ contain each source $s''$ of $G$ with $\ell(s'') = \{\min \ell\}$. Likewise, let $T_{\mathrm{may}} \subset V$ contain each sink $t'$ of $G$ with $\max \ell(t') = \max \ell$ and let $T_{\mathrm{must}} \subseteq T_{\mathrm{may}}$ contain each sink $t''$ of $G$ with $\ell(t'') = \{\max \ell\}$.

Construct a level assignment $\ell'$ from $\ell$ as follows. First, consider the case that $S_{\mathrm{must}}, T_{\mathrm{must}} \neq \emptyset$. Let $p$ denote the unique inclusion-minimal path that contains all vertices in $S_{\mathrm{must}}$ and no vertex in $T_{\mathrm{must}}$—if no such path exists there exists no separating level assignment, i.e., $G$ is not multilevel planar by Lemma 7. Set $\ell'(s) = \{\min \ell\}$ for all vertices $s \in S_{\mathrm{may}}$ that lie on $p$ and $\ell'(s) = \ell(s) \setminus \{\min \ell\}$ for all vertices $s \in S_{\mathrm{may}}$ that do not lie on $p$. Next, set $\ell'(t) = \ell(t) \setminus \{\max \ell\}$ for all vertices $t \in T_{\mathrm{may}}$ that lie on $p$. And set $\ell'(t) = \{\max \ell\}$ for all vertices $t \in T_{\mathrm{may}}$ that do not lie on $p$. Now consider the case $S_{\mathrm{must}} = \emptyset$. If $S_{\mathrm{must}} = \emptyset$, choose an arbitrary source $s \in S_{\mathrm{may}}$, define $p$ as the path that consists of only $s$ and proceed as in the previous case. The case $T_{\mathrm{must}} = \emptyset$ is symmetric by vertically mirroring. For each remaining vertex $v$ set $\ell'(v) = \ell(v)$. Finally, bring $\ell'$ into normal form.

We now show that $\ell'(v) \neq \emptyset$ for all $v \in V$. Non-empty intervals are explicitly assigned to all vertices in $S_{\mathrm{may}}$ and $T_{\mathrm{may}}$. We are left to show that bringing $\ell'$ into normal form does not create empty intervals. Changing the upper bound of a source's interval does not affect the intervals of that source's neighbors. The same applies to changing the lower bound of a sink's interval. Increasing the lower bound of a source's interval by one may increase the lower bound of the intervals of that source's neighbors (and, recursively, all vertices dominated by that source). However, because $\ell$ is in normal form this creates no empty intervals. Likewise, decreasing the upper bound of a sink's interval by one may decrease the upper bound of the intervals of all vertices that dominate that sink but creates no empty intervals. Moreover, there exists no vertex for whose interval both the lower bound and the upper bound is changed in this way. To see this, observe that the existence of such a vertex is equivalent to the existence of a directed path from a source $s$ with $\ell'(s) = \ell(s) \setminus \{\min \ell\}$, i.e., not on $p$ to a sink $t$ with $\ell'(t) = \ell(t) \setminus \{\max \ell\}$, i.e., on $p$. Such a directed path cannot exist because $p$ is delimited by sources $s_1, s_2$ with $\ell'(s_1) = \ell'(s_2) = \{\min \ell\}$ by construction.

Together with the fact that every level assignment that can be obtained from $\ell'$ is separating by construction and Lemma 7 we conclude the following.

**Theorem 3** *Let $G$ be an oriented cycle together with a multilevel assignment $\ell$. Then it can be decided in linear time whether $G$ admits a drawing that is multilevel planar with respect to $\ell$. Furthermore, if such a drawing exists, it can be computed within the same time.*

# 5 Hardness Results

Recall Theorem 1, which states that multilevel-planarity testing is in general NP-complete. Theorem 1 is a direct consequence of the fact that multilevel-planarity is a generalization of upward-planarity testing, which is known to be NP-complete [16]. We now show that multilevel-planarity testing is NP-complete even in very restricted cases, namely for $sT$-graphs without a fixed combinatorial embedding, for oriented trees and for embedded multi-source graphs with at most two possible levels for each vertex.

## 5.1 *sT*-Graphs with Variable Embedding

In Section 3, we showed that testing multilevel planarity of embedded $sT$-graphs is feasible in linear time, because for every inner sink there is a unique sink switch to cancel it with. We now

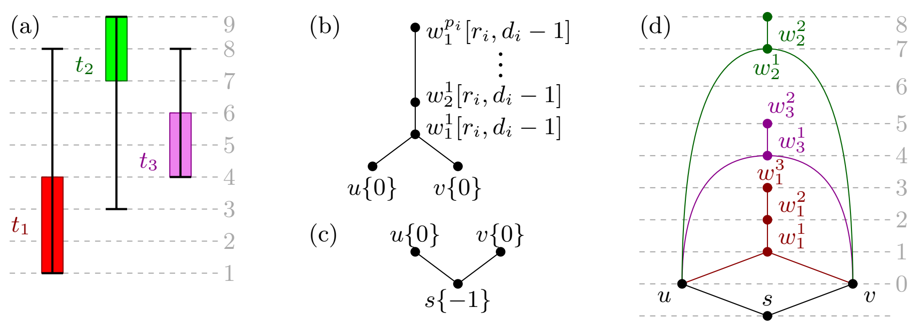


Figure 7: An instance of the Srtd problem that consists of the tasks $T = \{t_1, t_2, t_3\}$, the release times $r_1 = 1$, $r_2 = 3$, $r_3 = 4$, the deadlines $d_1 = 8$, $d_2 = 9$, $d_3 = 8$ and the processing times $p_1 = 3$, $p_2 = p_3 = 2$ (a). A task gadget (b) for each task and one base gadget (c) that provides the single source are used to turn the Srtd instance (a) into a multilevel-planarity testing instance (d).

show that dropping the requirement that the combinatorial embedding is fixed makes multilevel-planarity testing NP-hard. To this end, we reduce the Scheduling With Release Times And Deadlines (Srtd) problem, which is strongly NP-complete [15], to multilevel-planarity testing. An instance of this scheduling problem consists of a set of tasks $T = \{t_1, \ldots, t_n\}$ with individual release times $r_1, \ldots, r_n \in \mathbb{N}_+$, deadlines $d_1, \ldots, d_n \in \mathbb{N}_+$ and processing times $p_1, \ldots, p_n \in \mathbb{N}_+$ for each task so that $\sum_{i=1}^{n} p_i$ is bounded by a polynomial in $n$. See Figure 7 (a) for an example. The question is whether there is a non-preemptive schedule $\sigma : T \to \mathbb{N}$ that specifies the start time for each task, such that for each $i \in \{1, \ldots, n\}$ we get (1) $\sigma(t_i) \geq r_i$, i.e., no task starts before its release time, (2) $\sigma(t_i) + p_i \leq d_i$, i.e., each task finishes before its deadline, and (3) $\sigma(t_i) < \sigma(t_j) \Rightarrow \sigma(t_i) + p_i \leq \sigma(t_j)$ for any $j \in \{1, \ldots, n\} \setminus \{i\}$, i.e., no two tasks are executed at the same time.

Create for every task $t_i \in T$ a *task gadget* $\mathcal{T}_i$ that consists of two vertices $u$, $v$ together with a directed path $P_i = (w_i^1, w_i^2, \ldots, w_i^{p_i})$ of length $p_i - 1$; see Figure 7 (b). For each vertex $w_i^j$ on $P_i$ set $\ell(w_i^j) = [r_i, d_i - 1]$, i.e., all possible points of time at which this task can be executed. Set $\ell(u) = \ell(v) = \{0\}$. Join all task gadgets with a *base gadget*. The base gadget consists of three vertices $s, u, v$ and two edges $(s, u), (s, v)$, where $u$ is placed to the left of $v$; see Figure 7 (c). Set $\ell(s) = \{-1\}$ and, again, set $\ell(u) = \ell(v) = \{0\}$. Identify the common vertices $u$ and $v$ of all gadgets; see Figure 7 (d). Because Srtd is strongly NP-complete, the size of the resulting graph is polynomial in the size of the input. The idea of the construction is that because the task gadgets may not cross in a planar drawing and because their common vertices $u$ and $v$ are identified, they are stacked on top of each other, inducing a valid schedule of the associated tasks. Contrasting linear-time tests of upward planarity and level planarity for $sT$-graphs we show the following.

**Theorem 4** *Testing $sT$-graphs for multilevel planarity is NP-complete.*

**Proof:** Clearly, the problem is in NP. We reduce Srtd to multilevel-planarity testing. To this end, we show that the graph $G$ as described above is multilevel planar if and only if there is a valid one-processor schedule for the Srtd instance. To see this, start with a valid schedule $\sigma$. Define a level assignment $\gamma$ as follows. Start by setting $\gamma(s) = -1$, $\gamma(u) = \gamma(v) = 0$. And for $1 \leq i \leq n$ and $1 \leq j \leq p_i$, set $\gamma(w_i^j) = \sigma(t_i) + j$. Since $\sigma$ is non-preemptive, it induces a total order on the tasks, without loss of generality $\sigma(t_1) < \ldots < \sigma(t_n)$. Order the edges to the task gadgets $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$

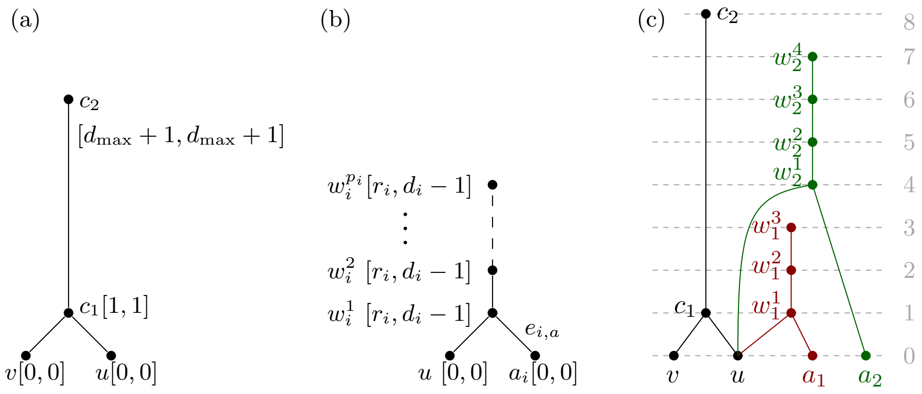


Figure 8: The base (a) and the task gadget (b) to transform a Srtd instance into a multilevel-planarity testing instance with $G$ being a tree. An example (c) with two task gadgets (in red and green).

from right to left at $u$, and from left to right at $v$. Observe that any sink of $G$ is the endpoint of a directed path of a task gadget. For $1 \le i < n$ cancel the sink $w_i^{p_i}$ by connecting it to $w_{i+1}^1$. This is possible because the schedule is valid. Then Lemma 6 implies that there exists a drawing of $G$ that is level-planar with respect to $\gamma$. Because it is $\gamma(v) \in \ell(v)$ for all $v \in V$ by construction, $G$ is multilevel planar with respect to $\ell$.

For the reverse direction, consider a drawing $\Gamma$ of $G$ that is multilevel planar with respect to $\ell$. Let $\gamma$ denote the level assignment induced by $\Gamma$. Further, let $\pi$ be the permutation of $\{1, 2, \dots, n\}$ so that the counter-clockwise order of edges around $u$ is $(s,u), (u, w_{\pi(1)}^1), (u, w_{\pi(2)}^1), \dots, (u, w_{\pi(n)}^1)$. For $1 \le i < n$ let $j = \pi(i)$ and $j' = \pi(i+1)$. The vertices $w_j^{p_j}$ and $w_{j'}^1$ are incident to a common face $f$. Note that $G$ is an $sT$-graph and because $s$ is not incident to $f$ it is an inner face, i.e., Property 2 of Lemma 3 applies. Because $w_j^{p_j}$ is incident to only one edge it is $\angle_{\Gamma,f}(w_j^{p_j}) > \pi$. Because $w_j^{p_j}$ and $w_{j'}^1$ are the only sink switches of $f$ it is $\angle_{\Gamma,f}(w_{j'}^1) < \pi$, i.e., $w_{j'}^1$ is the vertex with the greatest $y$-coordinate among all vertices incident to $f$. In particular, it is $\gamma(w_j^{p_j}) < \gamma(w_{j'}^1)$. For $1 \le i \le n$ and $j = \pi(i)$ set $\sigma(t_j) = \gamma(w_i^1)$. For $j = \pi(i)$ with $i < n$ it is $\sigma(t_j) + p_j < \sigma(t_{j'})$. Moreover, $\sigma(t_j) \ge r_j$ and $\sigma(t_j) + p_j \le d_j$ is ensured by the multilevel assignment. Hence, $\sigma$ is a valid schedule. □

## 5.2 Oriented Trees

We show NP-completeness of oriented trees with a very similar reduction as for $sT$-graphs without a fixed combinatorial embedding. As in Section 5.1, we reduce from Scheduling with Release Times and Deadlines, the required gadgets are only slightly different. Consider a Srtd instance $T = \{t_1, \dots, t_n\}$, $r_1, \dots, r_n$, $d_1, \dots, d_n$ and $p_1, \dots, p_n$, where $\sum_{i=1}^n p$ is bounded by a polynomial in $n$. Again we initialize $G$ with the base gadget shown in Figure 8 (a) and for each task we add one task gadget as shown in Figure 8 (b). The base gadget consists of two vertices $u$ and $v$ on level 0 both connected to one vertex $c_1$ on level 1, which in turn is connected to the final vertex $c_2$ on level $d_{\max}+1$, where $d_{\max} = \max_{i\in\{1,\dots,n\}} d_i$ is the maximum deadline among all tasks. The task gadget is the same as in the previous section, except that $v$ is replaced by a separate vertex $a_i$ per gadget. The base gadget and all task gadgets share one common vertex $u$, which is identified in $G$. The resulting graph $G$ is a tree and because Srtd is strongly NP-complete the size of $G$ is polynomial in the size of the Srtd instance.

**Theorem 5** *Testing oriented trees for multilevel planarity is NP-complete.*

**Proof:** The proof is very similar to the proof of Theorem 4. Clearly, the problem is in NP. Again, we reduce Srtd to multilevel-planarity testing. We show that the graph $G$ as described above is a multilevel planar if and only if there is a valid one-processor schedule for the Srtd instance. To see this, start with a valid schedule $\sigma$. Define a level assignment $\gamma$ as follows. Set $\gamma(u) = \gamma(v) = 0$ and for $i \in \{1, 2, \dots, n\}$ set $\gamma(a_i) = \gamma(b_i) = 0$. Moreover, set $\gamma(c_1) = 1$ and $\gamma(c_2) = d_{\max} + 1$. And for $1 \leq i \leq n$ and $1 \leq j \leq p_i$ set $\gamma(w_i^j) = \sigma(t_i) + j$. Since $\sigma$ is non-preemptive it induces a total order on the tasks, without loss of generality $\sigma(t_1) < \sigma(t_2) < \cdots < \sigma(t_n)$. Place $v, u, a_1, a_2, \dots, a_n$ in this order from left to right on level 0. Connect the edges incident to $u$ from left to right in the order $(u, c_1), (u, w_n^1), (u, w_{n-1}^1), \dots, (u, w_1^1)$. Observe that any sink of $G$ except for $c_2$ is the endpoint of a directed path of a task gadget. For $1 \leq i < n$ cancel the sink $w_i^{p_i}$ by connecting it to $w_{i+1}^1$. This is possible because the schedule is valid. Cancel the sink $w_n^{p_n}$ by connecting it to $c_2$. Create a new vertex $s$ and cancel all sources by connecting $s$ to them. Then Lemma 6 implies that there exists a drawing of $G$ that is level-planar with respect to $\gamma$. Since it is $\gamma(v) \in \ell(v)$ for all $v \in V$ by construction, $G$ is multilevel planar with respect to $\ell$.

For the reverse direction, consider a drawing $\Gamma$ of $G$ that is multilevel planar with respect to $\ell$. Obtain an $sT$-graph $G'$ together with a multilevel planar drawing $\Gamma'$ from $G$ and $\Gamma$ by adding a new vertex $s$ on level $-1$ and connecting it to $u, v$ and all $a_i$. Let $\gamma$ denote the level assignment induced by $\Gamma'$. Further, let $\pi$ be the permutation of $\{1, 2, \dots, n\}$ so that the counter-clockwise order of edges around $u$ is $(u, c_1), (u, w_{\pi(1)}^1), (u, w_{\pi(2)}^1), \dots, (u, w_{\pi(n)}^1)$. For $1 \leq i < n$ let $j = \pi(i)$ and $j' = \pi(i+1)$. The vertices $w_j^{p_j}$ and $w_{j'}^1$ are incident to a common face $f$ of $\Gamma'$. Because $\gamma(c_2) = \max \gamma$ the edge $(c_1, c_2)$ implies that all vertices $a_i$ for $1 \leq i \leq n$ lie on the same side of $u$ on level 0 and $v$ lies on the other side. This means that $f$ is an inner face. Together with the fact that $G'$ is an $sT$-graph this means that Property 2 of Lemma 3 applies. Because $w_j^{p_j}$ is incident to only one edge it is $\angle_{\Gamma',f}(w_j^{p_j}) > \pi$. Because $w_j^{p_j}$ and $w_{j'}^1$ are the only sink switches of $f$ it is $\angle_{\Gamma',f}(w_{j'}^1) < \pi$, i.e., $w_{j'}^1$ is the vertex with the greatest $y$-coordinate among all vertices incident to $f$. In particular, it is $\gamma(w_j^{p_j}) < \gamma(w_{j'}^1)$. For $1 \leq i \leq n$ and $j = \pi(i)$ set $\sigma(t_j) = \gamma(w_i^1)$. For $j = \pi(i)$ with $i < n$ it is $\sigma(t_j) + p_j < \sigma(t_{j'})$. Moreover, $\sigma(t_j) \geq r_j$ and $\sigma(t_j) + p_j \leq d_j$ is ensured by the multilevel assignment. Hence, $\sigma$ is a valid schedule. □

This also contrasts the results for upward planarity and level planarity, because every oriented tree is upward planar and all level graphs can be tested for level planarity in linear time.

## 5.3 Embedded Multi-Source Graphs

We show that multilevel-planarity testing for embedded graphs with multiple sources is NP-complete by reducing from planar monotone 3-Sat, which is known to be NP-complete [9]. An instance $\mathcal{I} = (\mathcal{V}, \mathcal{C}, \mathcal{E})$ of this problem consists of a set of variables $\mathcal{V}$, a set of clauses $\mathcal{C}$ and a planar drawing $\mathcal{E}$ of a so-called *variable-clause graph*, and it obeys the following restrictions. Each clause consists of at most three literals and it is *monotone*, i.e., it is either positive or negative, meaning that it consists of either only positive or only negative literals, respectively. We also assume that $\mathcal{C}$ contains at least one positive and at least one negative clause. The *variable-clause graph* consists of the nodes $\mathcal{V} \cup \mathcal{C}$. Two nodes are connected by an undirected arc if one of the nodes is a clause and the other node is a variable that appears as a literal in the clause. The drawing $\mathcal{E}$ of the

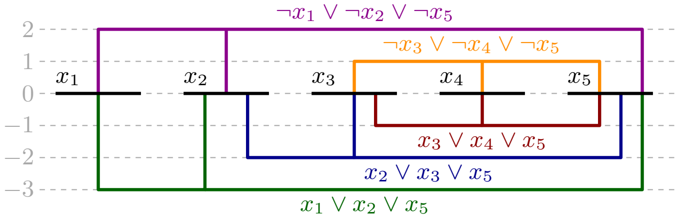


Figure 9: A planar monotone 3-Sat instance together with a rectilinear embedding of its variable-clause graph.

variable-clause graph is such that all variables lie on the horizontal straight line with $y$-coordinate 0, positive and negative clauses are drawn as horizontal line segments with integer $y$-coordinates below and above that line, respectively, and arcs connecting clauses and variables are drawn as non-intersecting vertical line segments; see Figure 9. We call $\mathcal{E}$ a *planar rectilinear drawing*.

Transform the variable-clause graph and its rectilinear drawing $\mathcal{E}$ into a multilevel graph by replacing each positive or negative clause with a positive or negative clause gadget and identifying common vertices (namely those vertices that are variables). The drawing $\mathcal{E}$ directly induces a combinatorial embedding and an outer face of the multilevel graph obtained in this way. Figure 10 (b) shows the gadget for a positive clause $(x_a \vee x_b \vee x_c)$. The vertices $x_a$, $x_b$ and $x_c$ are variables in $\mathcal{V}$. We call vertex $p_i$ the *pendulum*. A variable $x \in \mathcal{V}$ is set to true (false) if it lies on level 1 (level 0). In a positive clause gadget $p_i$ must lie on level 0. The idea is that it forces one variable to lie on level 1, i.e., be set to true. This is achieved by arranging the variables $x_a, x_b, x_c$ together with auxiliary vertices $v_{i,1}, v_{i,2}, v_{i,3}$ on a cycle $v_{i,1}, x_a, v_{i,2}, x_b, v_{i,3}, x_c, v_{i,1}$. This cycle encloses an inner face, the edge $(v_{i,1}, p_i)$ connects the pendulum to the cycle and the fixed embedding around $v_{i,1}$ ensures that the pendulum lies within this face. Let $y$ denote the $y$-coordinate of the clause $(x_a \vee x_b \vee x_c)$ in $\mathcal{E}$. Set $\ell(v_{i,1}) = \{2y\}$ and $\ell(v_{i,2} = \ell(v_{i,3}) = \{2y+1\}$. This ensures that the variable gadgets corresponding to distinct positive clauses do not intersect. The gadget for a negative clause $(\neg x_a \vee \neg x_b \vee \neg x_c)$ works symmetrically; the idea is that its pendulum forces one variable to lie on level 0, i.e., be set to false; see Figure 10 (a). To obtain the multilevel graph, replace each positive or negative clause with a positive or negative clause gadget and identify common vertices, namely those vertices that are variables. Figure 10 (c) shows the multilevel graph obtained from the planar monotone 3-Sat instance shown in Figure 9. In order for this graph to be multilevel planar, it must be possible to place the vertices that are variables on of the two possible levels so that all pendulums can be embedded within their gadgets, i.e., all clauses are satisfied.

**Theorem 6** *Testing embedded graphs for multilevel planarity is* NP*-complete, even when restricted to multilevel assignments $\ell$ with $|\ell(v)| \leq 2$ for each vertex $v$.*

**Proof:** Clearly, the problem is in NP. We reduce planar monotone 3-Sat to multilevel-planarity testing. To this end, we show that the multilevel graph $G$ derived from $(\mathcal{V}, \mathcal{C}, \mathcal{E})$ is multilevel planar if and only if $(\mathcal{V}, \mathcal{C}, \mathcal{E})$ is satisfiable.

Suppose that $\varphi$ is a satisfying truth assignment of the 3-Sat instance underlying $(\mathcal{V}, \mathcal{C}, \mathcal{E})$. Construct a drawing $\Gamma$ of $G$ that is multilevel planar with respect to $\ell$ by constructing a level assignment $\gamma$ as follows. Let $v \in \mathcal{V}$ be a variable. Recall that $v$ is a vertex in $G$. If $\varphi(v) = \text{true}$, set $\gamma(v) = 1$. Otherwise, set $\gamma(v) = 0$. Let $c_i \in \mathcal{C}$ be a positive clause. Draw the pendulum $p_i$ of $c_i$ below a vertex $v_i$ with $\varphi(v_i) = \text{true}$. Because $\varphi$ is a satisfying truth assignment such a $v_i$ exists. Now let $c_j \in \mathcal{C}$ be a negative clause. Draw the pendulum $p_j$ of $c_j$ above a vertex $v_j$

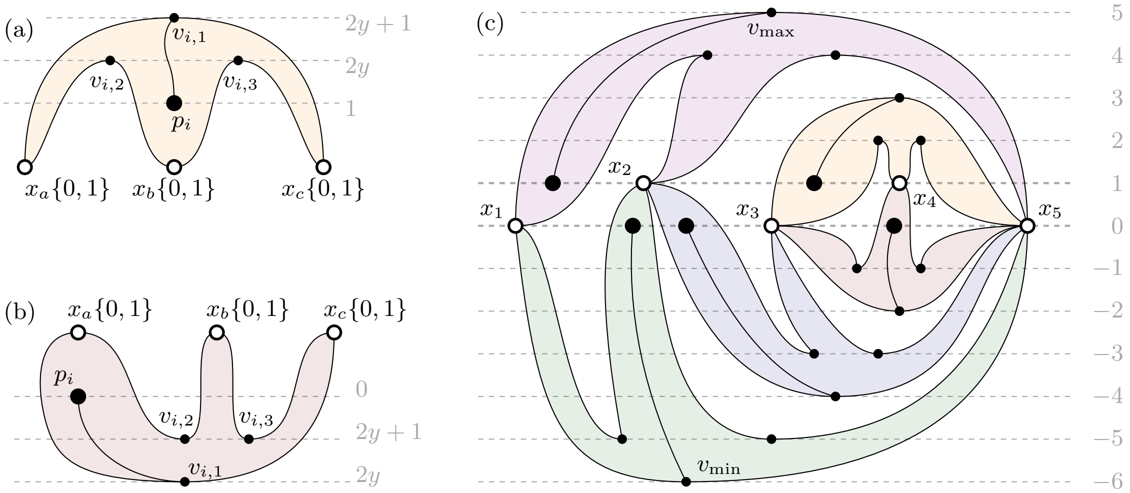


Figure 10: Gadgets for the clauses $(\neg x_a \vee \neg x_b \vee \neg x_c)$ (a) and $(x_a \vee x_b \vee x_c)$ (b). A multilevel-planar drawing of the graph constructed from the planar monotone 3-Sat instance shown in Figure 9 (c). The shaded faces correspond to the gadgets that substitute the clauses. In this multilevel-planar drawing, vertices $x_1$, $x_3$ and $x_5$ are on level 0, so variables $x_1 = x_3 = x_5 =$ false. On the other hand, vertices $x_2$ and $x_4$ are on level 1, so variables $x_2 = x_4 =$ true.

with $\varphi(v_j) =$ true. Since $c_j$ is a negative clause, a positive literal in $c_j$ corresponds to a variable set to false, and because $\varphi$ is a satisfying truth assignment such a $v_j$ exists. The resulting drawing is then level planar with respect to $\gamma$ and therefore multilevel planar with respect to $\ell$.

Now assume that $\Gamma$ is a combinatorially equivalent drawing of $G$ and is multilevel planar with respect to $\ell$. Due to the construction rules and because $\mathcal{C}$ contains at least one positive clause and one negative clause there exists exactly one face that is incident to both the vertex $v_{\min}$ with $\ell(v_{\min}) = \min \ell$ and the vertex $v_{\max}$ with $\ell(v_{\max}) = \max \ell$. This face must be the outer face; see Figure 10 (c). Let $\gamma$ denote the level assignment induced by $\Gamma$. Construct a truth assignment $\varphi$ as follows: Set the variable $v \in \mathcal{V}$ to true or false depending on whether it is $\gamma(v) = 1$ or $\gamma(v) = 0$, respectively. Because it is $\ell(v) = \{0, 1\}$, this always assigns a truth value to $v$. Consider the pendulum $p_i$ of a positive clause $c_i \in \mathcal{C}$. In a positive gadget, $p_i$ forces one of the variables in $c_i$, say $v_i$, to level 1, i.e., $\varphi(v_i) =$ true. Because $c_i$ is a positive clause it is then satisfied. In a negative gadget for a negative clause $c_j \in \mathcal{C}$, pendulum $p_j$ forces one of the variables in $c_j$, say $v_j$, to level 0, i.e., $\varphi(v_j) =$ false. Because $c_j$ is a negative clause, it is then satisfied. Hence, $\varphi$ is a satisfying truth assignment of $(\mathcal{V}, \mathcal{C}, \mathcal{E})$. □

# 6 Conclusion

We introduced and studied multilevel planarity, a generalization of both upward planarity and level planarity. We started by giving a linear-time algorithm to decide multilevel planarity of embedded $sT$-graphs. The correctness proof of this algorithm uses insights from both upward planarity and level planarity. In contrast to this, we showed that deciding the multilevel planarity of $sT$-graphs without a fixed embedding is NP-complete. We also gave a linear-time algorithm to decide multilevel planarity of oriented cycles, which is interesting because the existence of multiple sources makes many related problems NP-complete, e.g., testing upward planarity, partial level planarity or ordered level planarity. This positive result is contrasted by the fact that multilevel-planarity testing is NP-complete for oriented trees. Whether multilevel-planarity testing becomes tractable for trees with a given combinatorial embedding remains an open question. Deciding multilevel planarity remains NP-complete for embedded multi-source graphs where each vertex is assigned either to exactly one level, or to one of two adjacent levels. This contrasts the existence of efficient algorithms for testing upward planarity and level planarity of embedded multi-source graphs. The following table summarizes our results for multilevel planarity and relates them to existing results for upward planarity and level planarity.

| | fixed combinatorial embedding | | |
|---|---|---|---|
| | $st$-Graphs | $sT$-Graphs | arbitrary |
| Upward Planarity | $O(1)$ [6] | $O(n)$ [6] | P [6] |
| Multilevel Planarity | $O(1)$<br>(Corollary 1) | $O(n)$<br>(Theorem 2) | NPC<br>(Theorem 6) |
| Level Planarity | $O(1)$ [23] | $O(n)$ [23] | open |
| | not embedded | | |
| | Cycles | $sT$-Graphs | Trees |
| Upward Planarity | $O(n)$ [7] | $O(n)$ [7] | $O(1)$ [10] |
| Multilevel Planarity | $O(n)$<br>(Theorem 3) | NPC<br>(Theorem 4) | NPC<br>(Theorem 5) |
| Level Planarity | $O(n)$ [23] | $O(n)$ [23] | $O(n)$ [23] |

# References

[1] P. Angelini, G. Da Lozzo, G. Di Battista, F. Frati, M. Patrignani, and I. Rutter. Beyond level planarity: Cyclic, torus, and simultaneous level planarity. *Theor. Comput. Sci.*, 804:161–170, 2020. doi:10.1016/j.tcs.2019.11.024.

[2] P. Angelini, G. Da Lozzo, G. Di Battista, F. Frati, and V. Roselli. The importance of being proper: (in clustered-level planarity and $t$-level planarity). *Theor. Comput. Sci.*, 571:1–9, 2015. doi:10.1016/j.tcs.2014.12.019.

[3] P. Angelini, G. Di Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter. Testing planarity of partially embedded graphs. *ACM Trans. Algorithms*, 11(4):32:1–32:42, 2015. doi:10.1145/2629341.

[4] C. Bachmaier, F.-J. Brandenburg, and M. Forster. Radial level planarity testing and embedding in linear time. *J. Graph Algorithms Appl.*, 9(1):53–97, 2005. doi:10.7155/jgaa.00100.

[5] M. J. Bannister, W. E. Devanny, V. Dujmovic, D. Eppstein, and D. R. Wood. Track layouts, layered path decompositions, and leveled planarity. *Algorithmica*, 81(4):1561–1583, 2019. doi:10.1007/s00453-018-0487-5.

[6] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(6):476–497, 1994. doi:10.1007/BF01188716.

[7] P. Bertolazzi, G. Di Battista, C. Mannino, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM J. Comput.*, 27(1):132–169, 1998. doi:10.1137/S0097539794279626.

[8] G. Brückner and I. Rutter. Partial and constrained level planarity. In P. N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2000–2011. SIAM, 2017. doi:10.1137/1.9781611974782.130.

[9] M. de Berg and A. Khosravi. Optimal binary space partitions for segments in the plane. *Int. J. Comput. Geom. Appl.*, 22(3):187–206, 2012. doi:10.1142/S0218195912500045.

[10] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice-Hall, 1999.

[11] G. Di Battista and F. Frati. Efficient $c$-planarity testing for embedded flat clustered graphs with small faces. *J. Graph Algorithms Appl.*, 13(3):349–378, 2009. doi:10.7155/jgaa.00191.

[12] G. Di Battista and R. Tamassia. Algorithms for plane representations of acyclic digraphs. *Theor. Comput. Sci.*, 61:175–198, 1988. doi:10.1016/0304-3975(88)90123-5.

[13] W. Durant and A. Durant. *The Age of Louis XIV: The Story of Civilization.* Simon and Schuster, 2011.

[14] M. Forster and C. Bachmaier. Clustered level planarity. In P. van Emde Boas, J. Pokorný, M. Bieliková, and J. Stuller, editors, *SOFSEM 2004: Theory and Practice of Computer Science, 30th Conference on Current Trends in Theory and Practice of Computer Science, Merin, Czech Republic, January 24-30, 2004*, volume 2932 of *Lecture Notes in Computer Science*, pages 218–228. Springer, 2004. doi:10.1007/978-3-540-24618-3_18.

[15] M. R. Garey and D. S. Johnson. Two-processor scheduling with start-times and deadlines. *SIAM J. Comput.*, 6(3):416–426, 1977. doi:10.1137/0206029.

[16] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.

[17] M. Harrigan and P. Healy. Practical level planarity testing and layout with embedding constraints. In S. Hong, T. Nishizeki, and W. Quan, editors, *Graph Drawing, 15th International Symposium, GD 2007, Sydney, Australia, September 24-26, 2007. Revised Papers*, volume 4875 of *Lecture Notes in Computer Science*, pages 62–68. Springer, 2007. doi:10.1007/978-3-540-77537-9_9.

[18] P. Healy, A. Kuusik, and S. Leipert. A characterization of level planar graphs. *Discret. Math.*, 280(1-3):51–63, 2004. doi:10.1016/j.disc.2003.02.001.

[19] L. S. Heath and A. L. Rosenberg. Laying out graphs using queues. *SIAM J. Comput.*, 21(5):927–958, 1992. doi:10.1137/0221055.

[20] M. D. Hutton and A. Lubiw. Upward planar drawing of single-source acyclic digraphs. *SIAM J. Comput.*, 25(2):291–311, 1996. doi:10.1137/S0097539792235906.

[21] V. Jelínek, J. Kratochvíl, and I. Rutter. A kuratowski-type theorem for planarity of partially embedded graphs. *Comput. Geom.*, 46(4):466–492, 2013. doi:10.1016/j.comgeo.2012.07.005.

[22] E. Jelínková, J. Kára, J. Kratochvíl, M. Pergel, O. Suchý, and T. Vyskocil. Clustered planarity: Small clusters in cycles and eulerian graphs. *J. Graph Algorithms Appl.*, 13(3):379–422, 2009. doi:10.7155/jgaa.00192.

[23] M. Jünger and S. Leipert. Level planar embedding in linear time. *J. Graph Algorithms Appl.*, 6(1):67–113, 2002. doi:10.7155/jgaa.00045.

[24] H. Kamen. *Philip V of Spain: The King Who Reigned Twice.* Yale University Press, 2001.

[25] B. Klemz and G. Rote. Ordered level planarity and its relationship to geodesic planarity, bi-monotonicity, and variations of level planarity. *ACM Trans. Algorithms*, 15(4):53:1–53:25, 2019. doi:10.1145/3359587.

[26] S. Leipert. *Level Planarity Testing and Embedding in Linear Time.* PhD thesis, University of Cologne, 1998.

[27] A. Papakostas. Upward planarity testing of outerplanar dags. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing, DIMACS International Workshop, GD '94, Princeton, New Jersey, USA, October 10-12, 1994, Proceedings*, volume 894 of *Lecture Notes in Computer Science*, pages 298–306. Springer, 1994. doi:10.1007/3-540-58950-3_385.

[28] Wikipedia contributors. War of the Spanish Succession family tree template — Wikipedia, 2019. Online, accessed February 20th, 2019. URL: https://en.wikipedia.org/wiki/Template:War_of_the_Spanish_Succession_family_tree.