Journal of Computer and System Sciences 158 (2026) 103775

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

journal homepage: www.elsevier.com/locate/jcss

Rectilinear-upward planarity testing of digraphs ☆

Walter Didimo [a,*,1], Michael Kaufmann [b], Giuseppe Liotta [a,2], Giacomo Ortali [a,1], Maurizio Patrignani [c,2]

[a] University of Perugia, Perugia, Italy
[b] University of Tübingen, Tübingen, Germany
[c] Roma Tre University, Rome, Italy

A R T I C L E I N F O

Article history:
Received 26 July 2024
Received in revised form 1 October 2025
Accepted 19 January 2026
Available online 3 February 2026

Keywords:
Graph drawing
Orthogonal drawings
Upward drawings
Rectilinear planarity
Upward planarity
Parameterized complexity

A B S T R A C T

A *rectilinear-upward planar drawing* of a digraph G is a crossing-free drawing of G where each edge is either a horizontal or a vertical segment, and such that no directed edge points downward. Rectilinear-Upward Planarity Testing is the problem of deciding whether a digraph G admits a rectilinear-upward planar drawing. We study the complexity of Rectilinear-Upward Planarity Testing and provide several algorithmic results. Precisely, we prove that: (i) the problem is NP-complete, even if G is biconnected; (ii) it can be solved in linear time when an upward planar embedding of G is fixed; (iii) the problem is polynomial-time solvable for biconnected digraphs of treewidth at most two, i.e., for digraphs whose underlying undirected graph is a series-parallel graph; (iv) the problem is fixed-parameter tractable (namely, fixed-parameter linear) for all biconnected graphs, when parameterized by the number of sources and sinks in the digraph.

© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

A *rectilinear planar drawing* of a graph G is a crossing-free drawing of G where vertices are placed at distinct points in the plane (possibly at grid points) and edges are drawn as either horizontal segments or vertical segments. Rectilinear Planarity Testing is the problem of deciding whether a planar graph admits a rectilinear planar drawing. Besides the theoretical beauty of the problem, which belongs to the vast literature about graph planarity testing (see, e.g., [1–3] for books and surveys), the question is at the heart of those technologies that display networked data by means of orthogonal layouts, which find applications in a variety of fields, from software engineering to bioinformatics, from data bases to computer networks (see, e.g., [4,5]).

☆ A preliminary version of this research appeared in the proceedings of the 34th International Symposium on Algorithms and Computation (ISAAC 2023). With respect to the preliminary version, this extended version contains full technical details and substantial improvements of the time complexity of the algorithms in Section 6.

* Corresponding author.
E-mail addresses: walter.didimo@unipg.it (W. Didimo), mk@informatik.uni-tuebingen.de (M. Kaufmann), giuseppe.liotta@unipg.it (G. Liotta), giacomo.ortali@unipg.it (G. Ortali), maurizio.patrignani@uniroma3.it (M. Patrignani).

1 Work partially supported by Project AIDMIX—Artificial Intelligence for Decision Making: Methods for Interpretability and eXplainability - Ricerca di Base 2021 - University of Perugia.
2 Work partially supported by MUR of Italy, PRIN Project n. 2022TS4Y3N - EXPAND: scalable algorithms for EXPloratory Analyses of heterogeneous and dynamic Networked Data.

https://doi.org/10.1016/j.jcss.2026.103775
0022-0000/© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

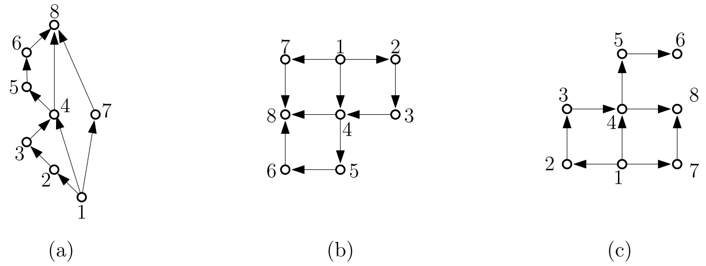


Fig. 1. A graph G that is upward planar, rectilinear planar, but not rectilinear-upward planar. (a) An upward planar drawing of G. (b) A rectilinear planar drawing of G. (c) A rectilinear-upward planar drawing of G without edge $(6, 8)$.

Rectilinear Planarity Testing is known to be NP-complete [6]. However, polynomial-time solutions are given both for constrained versions of the problem and for restricted families of graphs. Namely, Rectilinear Planarity Testing can be solved in polynomial time in the so-called *fixed-embedding setting*, that is, when the input is given with a planar embedding and the testing algorithm is not allowed to change the embedding (see, e.g., [7]). Also, polynomial-time solutions are known for graphs of bounded treewidth and for sub-cubic graphs (see, e.g., [8–15]).

In this paper we study rectilinear planar drawings of directed graphs (digraphs). We want to test whether a digraph G admits a rectilinear planar drawing with the additional constraint that no directed edge points downward. We call such a drawing a *rectilinear-upward planar drawing* and the testing problem Rectilinear-Upward Planarity Testing. It may be worth recalling that the problem of testing whether a digraph admits an *upward planar drawing*, i.e., a planar drawing where each edge is monotonically increasing in the upward direction according to its orientation, is NP-complete [6]. See also [16–23] for polynomial-time solutions and parameterized approaches on restricted graph families or scenarios. Fig. 1 shows an example of a digraph that admits both an upward planar drawing and a rectilinear planar drawing, but that does not admit a rectilinear-upward planar drawing. Our results are the following.

- We prove that Rectilinear-Upward Planarity Testing is NP-complete, even if the input digraph is biconnected (Section 4). The proof exploits the reduction from a new NP-hard graph orientation problem that we introduce in this paper and that can be considered of independent interest.
- We show that Rectilinear-Upward Planarity Testing can be solved in linear-time when an upward planar embedding of G is fixed as part of the input (Section 5). We remark that both the problem of testing rectilinear planarity and of testing upward planarity in linear time in the fixed-embedding setting are among of the most famous and long-standing open problems in graph drawing (see, e.g., [24,25]).
- We consider the variable-embedding setting, where the algorithm is free to choose the planar embedding of the input graph, and we focus on biconnected digraphs (Section 6). We show that Rectilinear-Upward Planarity Testing can be solved in quadratic time for digraphs with treewidth at most two, i.e., when the underlying undirected graph is series-parallel. It may worth recalling that polynomial-time algorithms for treewidth-2 graphs are known for upward planarity testing without the rectilinearity constraint and for rectilinear planarity testing without the upwardness constraint (see, e.g., [26–28]). We also show that Rectilinear-Upward Planarity Testing is fixed-parameter linear (FPL) when parameterized by the number k of sources and sinks of the digraph. Namely, for any n-vertex digraph our testing algorithm is single-exponential in k and has a linear factor in n. We remark that parameterized complexities of upward and rectilinear planarity testing are topics that have been receiving increasing attention (see, e.g., [20,29,30]).

From a technical point of view, in Section 4 we introduce the 1-2-Switch-Flow problem, which we show is NP-hard, and then reduce Rectilinear-Upward Planarity Testing from 1-2-Switch-Flow. When the upward planar embedding is fixed, we prove in Section 5 that Rectilinear-Upward Planarity Testing can be solved in linear-time by means of a 2-SAT formulation, instead of using network-flow models as done in the standard approaches for testing both rectilinear and upward planarity (see, e.g., [16,17,31,32,7]). As for the results of Section 6 in which the embedding is not fixed, we rely on the concept of *rectilinear-upward spirality*. It combines the notion of spirality introduced in [8] to measure how much a triconnected component of a rectilinear drawing can be "rolled up", with additional information about the orientation of the edges incident to the poles of the triconnected components.

2. Basic definitions and properties

For basic definitions on graph drawing and planarity refer to [1]. We assume to work with connected graphs, as otherwise we can treat each connected component of the graph independently. A 4-*graph* is a graph with vertex-degree at most four.

Planar graphs and drawings. Let G be a graph. A *drawing* Γ of G maps each vertex to a distinct point of the plane and each edge to a Jordan arc between its endpoints. A drawing Γ is *planar* if no two edges of G cross in Γ. A graph is *planar* if it admits a planar drawing. A planar drawing Γ of G divides the plane into topologically connected regions called *faces*. Exactly one of these regions is an infinite region, called the *external face*. The other regions are the *internal faces*. The set of (internal and external) faces determined by a planar drawing of G is a *planar embedding* of G. Such an embedding can be described by listing for each face f the clockwise circular sequence of vertices and edges on the boundary of f, and by the indication of what face is the external one. Equivalently, a planar embedding of G can be described by a *rotation system*, i.e., the circular clockwise order of the edges incident to each vertex and the choice of an external face. A planar graph G together with a planar embedding is an *embedded planar graph* or simply a *plane graph*. An *embedding-preserving drawing* of a plane graph G is a drawing that induces the same planar embedding of G.

Upward planar drawings. In an *upward drawing* of a digraph G each edge is represented as a Jordan arc monotonically increasing in the upward direction, according to its orientation; see Fig. 1a. A digraph G is *upward planar* if it admits an upward planar drawing. Clearly, a necessary (but not sufficient) condition for G to be upward planar is that G is acyclic.

Orthogonal drawings and representations. Let G be a planar (undirected or directed) 4-graph, and let Γ be a planar drawing of G. We say that Γ is an *orthogonal planar drawing* of G if each edge is drawn as a sequence of horizontal and vertical segments. A *bend* on an edge is the contact point between a horizontal and a vertical segment of the edge. An *orthogonal planar representation* H of a planar graph G is a class of shape equivalent orthogonal planar drawings; namely, H describes the planar embedding of G, the sequence of left/right bends along the edges, and the angles at every vertex of G, each angle formed by two (possibly coincident) consecutive edges around the vertex and expressed as a value in the set $\{90^\circ, 180^\circ, 270^\circ, 360^\circ\}$. If H is the orthogonal representation of an orthogonal planar drawing Γ, we also say that Γ *preserves* H and that Γ is a *drawing of* H. A drawing of H can be computed in linear time [7], thus we can concentrate on computing orthogonal representations rather than drawings.

Orthogonal planar drawings (resp. representations) without bends are called *rectilinear planar drawings* (resp. *rectilinear planar representations*) [1]; see Fig. 1b. A graph G is *rectilinear planar* if it admits a rectilinear planar drawing (or representation). We say that a rectilinear planar representation H is *oriented* if it also specifies for each edge (u, v) of G the relative position of u with respect to v, i.e., whether u must be to the left, to the right, above, or below v in every rectilinear drawing of H; in particular, this information establishes for each edge e of G if e is horizontal or vertical in H (it is actually enough to specify the relative position of the end-vertices of one edge of H to establish the relative position of the end-vertices for every other edge). Note that the definition of oriented rectilinear representation H of G has nothing to do with the orientation of the edges when G is a digraph. For any rectilinear representation of G there are four different oriented versions of it, obtained by rotating one of them by an angle of $k \cdot 90^\circ$, for $k = 0, 1, 2, 3$.

Rectilinear-upward planar representations In this paper we deal with drawings of digraphs that are at the same time rectilinear and upward. More precisely, we do not require that an edge is strictly upward (which would prevent us from drawing it as a horizontal segment), but rather we exclude that it is drawn downward. Formally, let G be an acyclic planar 4-digraph and let Γ be a planar drawing of G. We say that Γ is a *rectilinear-upward planar drawing* of G if Γ is a rectilinear planar drawing of G with no directed edge that points downward; see, e.g., Fig. 1c. This corresponds to saying that a rectilinear-upward planar drawing Γ induces an oriented rectilinear planar representation H of G with the property that for each directed edge (u, v) of G, vertex u is never above vertex v. We say that H is a *rectilinear-upward planar representation* of G. As for rectilinear representations, H describes a class of shape equivalent rectilinear-upward planar drawings. A digraph G is *rectilinear-upward planar* if it admits a rectilinear-upward planar drawing, or equivalently, a rectilinear-upward planar representation. Clearly, rectilinear planarity is necessary for rectilinear-upward planarity. The next property implies that also upward planarity is necessary for rectilinear-upward planarity. As already observed, both rectilinear planarity and upward planarity are not sufficient conditions when considered independently (see Fig. 1).

Property 1. *If Γ is a rectilinear-upward planar drawing of a digraph G, then Γ can be transformed into an upward planar drawing of G with the same planar embedding as Γ.*

Proof. We describe a simple algorithm that transforms Γ into an upward planar drawing of G, while preserving the planar embedding (refer to Fig. 2 for an illustration).

- *Step 1.* Temporarily remove from Γ the vertical edges. The resulting drawing is a collection of horizontal paths.
- *Step 2.* For each path P temporarily smooth all degree-2 vertices that are neither sources nor sinks in P, i.e., those vertices that have both an incoming edge and an outgoing edge. Namely, if v is such a degree-2 vertex and if (u, v) and (v, w) are its incident directed edges, smoothing v means that we remove (u, v) and (v, w) and add a directed edge (u, w). At this point the new drawing has only sources, sinks or isolated vertices.
- *Step 3.* For each source v in the drawing, decrease the y-coordinate of v by an infinitesimal amount $\varepsilon > 0$; in particular, ε must be smaller than half the vertical distance between any two horizontal paths of the drawing. Analogously, increase the y-coordinate of each sink of the drawing by ε.
- *Step 4.* Reinsert each smoothed vertex at a position that is the vertical projection of its original position on the line corresponding to the path to which v belonged.
- *Step 5.* Reinsert the vertical edges between their original end-vertices. Note that these edges can be in some cases shorter or longer than in Γ.

It is immediate to verify that the new drawing is an upward planar drawing of G that has the same embedding as Γ. □

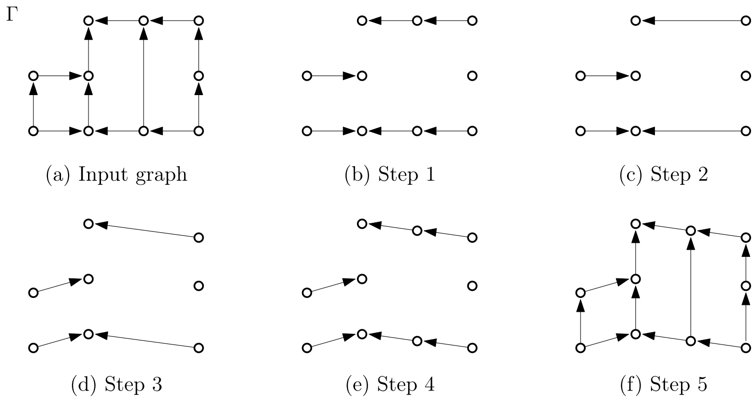


Fig. 2. Illustration of the algorithm that transforms a rectilinear-upward planar drawing into an upward planar drawing.

Table 1
Summary of our results for the Rectilinear-Upward Planarity Testing problem. For each setting we report the time complexity of the problem and the reference to the corresponding theorem.

Setting (type of input graph)	Time Complexity	Reference
General planar graphs	NP-complete	Theorem 1
Graphs with fixed upward planar embedding	$O(n)$	Theorem 3
Series-parallel digraphs	$O(n^2)$	Theorem 5
Biconnected planar digraphs with k switches	$O(2^{k\log k+2k}\cdot n)$	Theorem 6

In the remainder we only consider planar 4-digraphs, thus we often omit the term “planar” and we avoid to specify that the vertex-degree is at most four. Also, we often use the abbreviation “RU” in place of “rectilinear-upward”. Finally, we implicitly assume that the input digraphs are acyclic, as otherwise an upward drawing, and hence an RU drawing, cannot exist. Acyclicity can be tested in linear time, by a classical depth first search.

In any RU representation H of a digraph G each vertex v is *bimodal*, i.e., all the incoming edges of v (as well as all the outgoing edges of v) are consecutive around v. More specifically, H induces: (a) a planar embedding of G and, (b) for each vertex v of G, a linear left-to-right (possibly empty) list of the incoming edges of v and a linear left-to-right (possibly empty) list of the outgoing edges of v. The information (a) and (b) together are called an *upward planar embedding* of G. A digraph G is *upward plane* if it comes with a given upward planar embedding. An RU representation of an upward plane digraph G (if any) is an RU representation of G that preserves its upward planar embedding. Note that, given information (a) and (b) for a planar digraph G, it can be easily checked in linear time whether this pair correctly defines an upward planar embedding, i.e., if there exists an upward planar drawing of G whose upward planar embedding coincides with (a) and (b) (see e.g. [17]).

3. Overview of the results

Before entering into the technical details of our contributions, we summarize below our main results and give references to the related theorems. See also Table 1.

- **Result 1.** Rectilinear-Upward Planarity Testing is NP-complete, even for biconnected digraphs; see Theorem 1.
- **Result 2.** Given a planar digraph G with n vertices and with a fixed upward planar embedding, the Rectilinear-Upward Planarity Testing problem for G can be solved in $O(n)$ time; see Theorem 3.
- **Result 3.** Given a series-parallel digraph G with n vertices, Rectilinear-Upward Planarity Testing for G can be solved in $O(n^2)$ time; see Theorem 5.
- **Result 4.** Given a biconnected planar digraph G with n vertices and k switches (sources and sinks), Rectilinear-Upward Planarity Testing problem for G is fixed-parameter linear with parameter k and complexity $O(2^{k\log k+2k}\cdot n)$ time; see Theorem 6.

4. NP-completeness of Rectilinear-Upward Planarity Testing

To prove the hardness of Rectilinear-Upward Planarity Testing, we use a reduction from the 1-2-Switch-Flow problem, that is introduced hereunder and that may be considered of independent interest. Informally, this problem asks for an orientation of the edges of an edge-labeled planar undirected graph such that, when regarding the labels as flows, we have

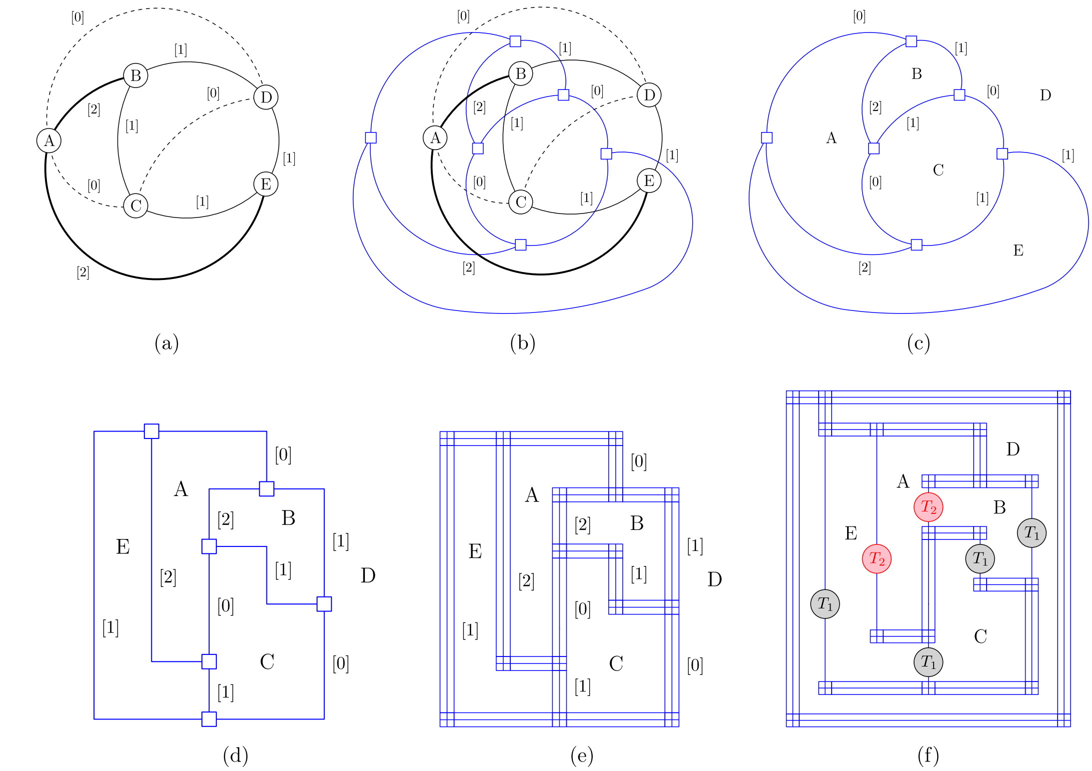


Fig. 3. The five steps of the reduction from 12SW to rectilinear-upward planarity testing. (a) **Step 1**: Graph G^+. (b) and (c) **Step 2**: Graph G^*, dual of G^+. (d) **Step 3**: Orthogonal drawing Γ_{G^*}. (e) **Step 4**: The positive instance F of rectilinear upward planarity testing. (f) **Step 5**: Instance I_{RUPT}. The symbols T_1 and T_2 represent the tendril subgraphs depicted in Fig. 6a and Fig. 6b, respectively. A rectilinear-upward planar representation of instance I_{RUPT}, corresponding to a solution of the 12SW instance, is provided in Fig. 8c. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

a feasible flow for the oriented graph. In other words, the orientation of the edges must yield flow conservation at each vertex. If we call [k]-edge an edge whose label is k, in the 1-2-Switch-Flow problem we only have [1]- and [2]-edges. The hardness of 1-2-Switch-Flow is proved in Section 4.1, with a reduction from the problem Flow Orientation, which allows for arbitrary edge labels and is shown to be NP-complete in [33] even if edge labels are $O(\sqrt{n})$, where n is the number of vertices of the graph.

Problem: 1-2-Switch-Flow (12SW)

Instance: A planar undirected graph $G = (V, E)$ where each edge $e \in E$ is labeled with a value $f_e \in \{1, 2\}$.

Question: Does there exist an orientation for the edges in E such that for each vertex the sum of the values of the incoming edges equals the sum of values of the outgoing edges?

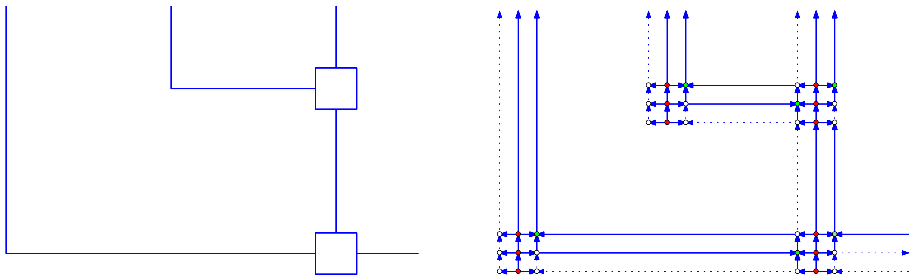
Let $G = (V, E)$ be an instance of 1-2-Switch-Flow. We build an equivalent instance of rectilinear-upward planarity testing as follows.

Step 1. Compute a planar embedding of G, for example with the algorithm in [34], and planarly add extra [0]-edges in such a way to obtain a maximal plane graph G^+. Observe that, since the inserted dummy edges have flow value 0, G^+ admits a feasible flow if and only if G does. Fig. 3a shows an example of maximal planar instance.

Step 2. Compute the dual plane graph G^* of G^+. Graph G^* has a vertex for each (triangular) face of G^+ and has an edge between two vertices if and only if the two corresponding faces of G^+ are adjacent. Label the edges of G^* with the flow value of the corresponding edge of G^+. Figs. 3b and 3c show the dual graph of the maximal planar instance depicted in Fig. 3a.

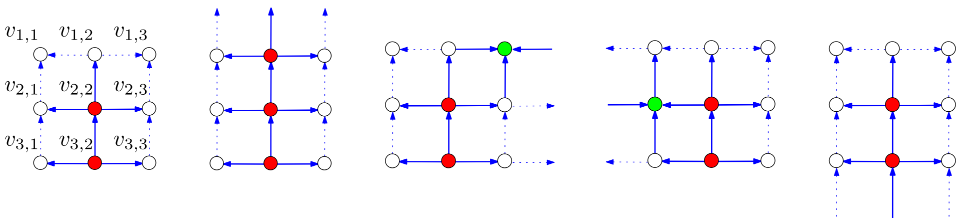
Step 3. Compute an orthogonal drawing Γ_{G^*} of G^* with the algorithm in [35] that takes as input a plane biconnected graph of vertex-degree at most 4, and computes an orthogonal drawing with at most $2n + 4$ bends such that each edge has at least one vertical segment. Fig. 3d shows such an orthogonal drawing for the dual graph of Fig. 3c.

Step 4. Transform Γ_{G^*} into a positive instance F of rectilinear-upward planarity testing by replacing horizontal and vertical segments with suitable graphs. In particular, each edge segment of Γ_{G^*} is replaced by three parallel edges



(a) A portion of the drawing Γ_{G^*} (b) A portion of instance F

Fig. 4. The construction of instance F. Vertices filled red have three outgoing edges. Vertices filled green have three incoming edges. Thick edges are incident to a red or green vertex (or both).



(a) (b) (c) (d) (e)

Fig. 5. A figure for the proof of Lemma 1.

and each vertex or bend of Γ_{G^*} is replaced by a 3×3 grid (see Fig. 3e and Fig. 4). Regarding orientation: all vertical segments are oriented towards the top. Horizontal segments are oriented depending on their placement: if they are part of a 3×3 grid they are oriented away from the center of the grid. Otherwise, if they are replacing a horizontal segment of Γ_{G^*}, then the middle one is oriented towards the right and the top and bottom ones are oriented towards the left (see Fig. 4b). The instance F of rectilinear-upward planarity testing has maximum vertex-degree four and, as it will be proved by Lemma 1, it has a unique rectilinear-upward planar representation $\mathcal{H}_F$ up to a horizontal flip.

Step 5. Starting from $\mathcal{H}_F$, build the desired instance I_{RUPT} of rectilinear-upward planarity. First, for each [1]-edge e of G^* ([2]-edge e, respectively) identify a rectangular box of F corresponding to a vertical segment of Γ_{G^*} and replace it with the subgraph T_1 depicted in Fig. 6a (the subgraph T_2 depicted in Fig. 6b, respectively), called *tendril*. Finally, add a framework all around I_{RUPT} and connect it to a vertex or bend in the upper side of Γ_{G^*} as shown in Fig. 3f in order to obtain an instance of rectilinear-upward planarity testing where the external face is constrained to be rectangular and to be the external cycle of such a framework.

Lemma 1. *Starting from any instance of* 1-2-Switch-Flow*, the instance F of rectilinear-upward planarity testing obtained at Step 4 of the above described reduction admits a unique rectilinear-upward planar representation up to a horizontal flip.*

Proof. Intuitively, the proof is based on the fact that a high number of vertices of F have either three incoming edges or three outgoing edges. Therefore, the direction of the edges incident to these vertices is fixed up to a horizontal flip. The few edges of F whose direction is not determined by the above consideration have a direction determined by their adjacent edges. More formally, consider a 3×3 grid of F that replaces a vertex or a bend of Γ_{G^*} (refer to Fig. 5a). Denote the vertices of the grid $v_{i,j}$, with $i = 1, 2, 3$ counting the rows of the grid from top to bottom and $j = 1, 2, 3$ counting the columns of the grid from left to right. Since the vertical edges of the grid are oriented towards the top and the horizontal edges are oriented away from the center of the grid, $v_{2,2}$ and $v_{3,2}$ have three outgoing edges (red vertices of Fig. 5a). Now, consider the case that the bend or vertex of Γ_{G^*} that is represented by the 3×3 grid is connected to a segment from above. By construction the 3×3 grid of F has three edges leaving from the top row and $v_{1,2}$ has three outgoing edges (refer to Fig. 5b). If instead it is connected to the right, then $v_{1,3}$ has three incoming edges (refer to Fig. 5c). If it is connected from the left, then $v_{2,1}$ has three incoming edges (refer to Fig. 5d). The configuration when the 3×3 grid is connected from the bottom is depicted in Fig. 5e. In any of the above considered cases one of the three edges of F that represent the segment that is incident to the bend or vertex of Γ_{G^*} is incident to a vertex of F whose embedding is fixed (thick edges of Fig. 5). Therefore, the *fixed subgraph* of F induced by the edges incident to vertices with fixed rectilinear representation (thick edges of Fig. 5) is connected and reaches each 3×3 grid of F. Since G^* is biconnected and has no degree-two vertex the fixed subgraph admits a unique rectilinear-upward planar representation up to a horizontal flip. It can be easily seen

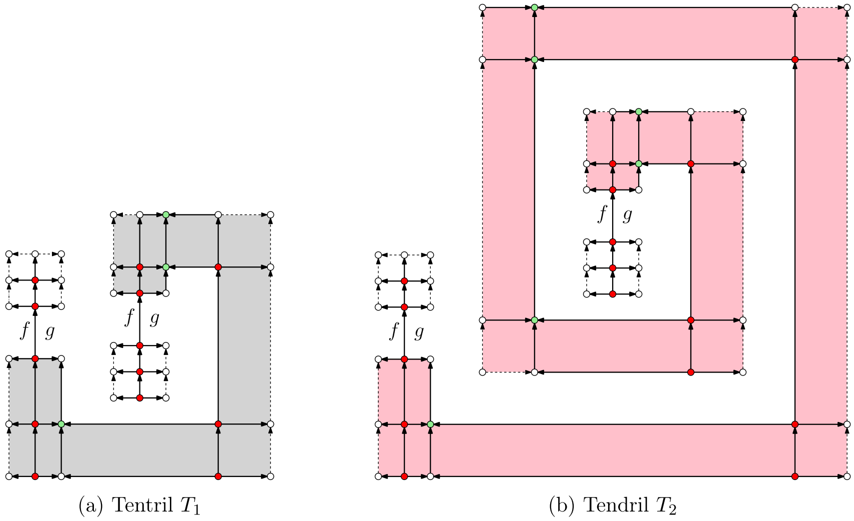


Fig. 6. Tendrils T_1 (a) and T_2 (b). Red vertices have three outgoing edges. Green vertices have three incoming edges. Solid edges are incident either to a red or to a green vertex (or both).

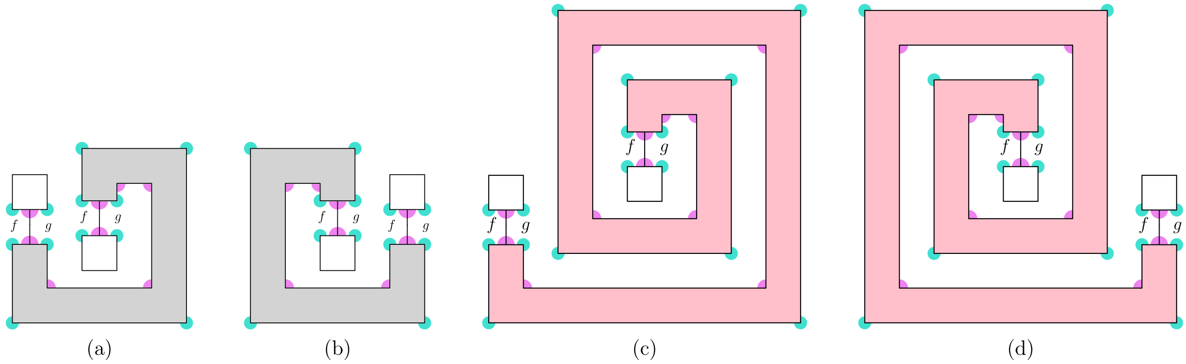


Fig. 7. The two possible RU representations of tendril T_1 (a) and (b) and of tendril T_2 (c) and (d). Highlighted in green (purple, respectively) are 270° angles (90° angles, respectively).

that the remaining edges (dotted edges of Fig. 5) admit a unique orthogonal representation once the fixed subgraph has been drawn. □

Lemma 2. *The tendrils depicted in Fig. 6 admit a unique RU representation up to a horizontal flip.*

Proof. Consider first tendril T_1 depicted in Fig. 6a. Observe that by the orientation of the edges of T_1 some vertices have three outgoing edges (filled red vertices of Fig. 6a) and some other have three incoming edges (filled green vertices of Fig. 6a). The edges incident to these vertices (thick solid edges of Fig. 6a) have a unique RU representation up to a horizontal flip. We call *fixed subgraph* the subgraph of T_1 induced by the edges incident to such vertices. As it can be seen from Fig. 6a the fixed subgraph traverses the whole tendril. Once one of the two possible horizontal flips for the fixed subgraph is chosen, the remaining undecided edges (dashed edges of Fig. 6a) also have a unique representation determined by the smallest facial cycle of F traversing the edge. With analogous reasoning, it can be shown that tendril T_2 depicted in Fig. 6b admits a unique RU representation up to a horizontal flip. □

Lemma 3. *Depending on the horizontal flip of the RU representation of tendril T_1 (of tendril T_2, respectively) one of the two statements below hold for the faces f and g on the left and on the right of T_1 (of T_2, respectively):*

- *the boundary of f makes eight 270° angles and four 90° angles (twelve 270° angles and four 90° angles, respectively) while the boundary of g makes four 270° angles and eight 90° angles (four 270° angles and twelve 90° angles, respectively).*
- *the boundary of f makes four 270° angles and eight 90° angles (four 270° angles and twelve 90° angles, respectively) while the boundary of g makes eight 270° angles and four 90° angles (twelve 270° angles and four 90° angles, respectively).*

Proof. By Lemma 2 in any RU representation of G tendril T_1 either has the RU representation depicted in Fig. 7a or the one depicted in Fig. 7b. In the RU representation of tendril T_1 depicted in Fig. 7a traversing face f eight 270° angles and four 90° angles are encountered, while traversing face g four 270° angles and eight 90° angles are encountered. Conversely, in the RU representation depicted in Fig. 7b traversing face f four 270° angles and eight 90° angles are encountered, while traversing face g eight 270° angles and four 90° angles are encountered.

Analogously, in any RU representation of G tendril T_2 either has the RU representation depicted in Fig. 7c or the one depicted in Fig. 7d. In the RU representation of tendril T_2 depicted in Fig. 7c traversing face f twelve 270° angles and four 90° angles are encountered, while traversing face g four 270° angles and twelve 90° angles are encountered. Conversely, in the RU representation depicted in Fig. 7d traversing face f four 270° angles and twelve 90° angles are encountered, while traversing face g twelve 270° angles and four 90° angles are encountered. □

Theorem 1. Rectilinear-Upward Planarity Testing *is NP-complete.*

Proof. The problem is in NP since the recognition of an RU representation is polynomial. As for the hardness, we show that the above described reduction from 1-2-Switch-Flow constructs an instance I_{RUPT} of rectilinear-upward planarity testing that admits an RU representation if and only if its tendrils are embedded in such a way that, for each face of Γ_{G^*}, the number of extra 270° angles provided by some tendrils equals the number of extra 90° angles provided by the remaining tendrils of the same face and this is equivalent to finding a feasible flow for the original 1-2-Switch-Flow instance.

Let $\Gamma_{I_{\text{RUPT}}}$ be an RU representation of I_{RUPT}. Observe that, the external face of $\Gamma_{I_{\text{RUPT}}}$ is necessarily the rectangular face that corresponds to the external cycle of the framework added to F to obtain I_{RUPT}. By [36], traversing any internal face h of $\Gamma_{I_{\text{RUPT}}}$ one encounters a number n_{90} of 90° angles and a number n_{270} of 270° angles such that $n_{90} - n_{270} = 4$. Observe that the above statement trivially holds for all the internal faces of I_{RUPT} that are rectangular. The internal faces of I_{RUPT} that are not rectangular correspond to the faces of Γ_{G^*}. With respect to the RU representation that they have in F, these faces contain the additional 90° and 270° angles stated by Lemma 3. Therefore, we can interpret:

- the RU representation of T_1 depicted in Fig. 7a as an injection of one unit of flow, represented by four extra 270° angles, from g to f;
- the RU representation of T_1 depicted in Fig. 7b as an injection of one unit of flow from f to g;
- the RU representation of T_2 depicted in Fig. 7c as an injection of two units of flow, represented by eight extra 270° angles, from g to f; and
- the RU representation of T_2 depicted in Fig. 7d as an injection of two units of flow from f to g.

Since in order to satisfy the conditions in [36] the number of 90° angles injected into each face by its incident tendrils is equal to the number of injected 270° angles, we have that the flow balance of the corresponding vertex of G is zero. It follows that the RU representation $\Gamma_{I_{\text{RUPT}}}$ of I_{RUPT} provides a feasible flow for the original 1-2-Switch-Flow instance.

Conversely, suppose that the original 1-2-Switch-Flow instance G is a positive instance and consider a feasible flow for G (see Fig. 8a). If one unit of flow transits from face g to face f, use the RU representation depicted in Fig. 7a for the tendril T_1 that separates g from f. If, instead, one unit of flow transits from f to g, use the RU representation depicted in Fig. 7b. If two units of flow transit from g to f, use the RU representation depicted in Fig. 7c for the tendril T_2 that separates g from f. Finally, if two units of flow transit from f to g use the RU representation depicted in Fig. 7d (see Fig. 8b). Since the flow of each vertex of G is balanced, the number of extra 90° angles injected into each face by its tendrils equals the number of injected 270° angles and, by [36], the obtained embedding for I_{RUPT} admits an RU representation (see Fig. 8c). □

4.1. *NP-completeness of* 1-2-Switch-Flow

In order to show that 1-2-Switch-Flow is NP-hard, we reduce the following NP-complete problem:

Problem: Flow Orientation (FO)

Instance: A planar undirected graph $G = (V, E)$ where each edge $e \in E$ is labeled with a flow value f_e.

Question: Does there exist an orientation for the edges in E such that each vertex has a balanced flow? A vertex has a *balanced* flow if the total flow entering it from the incoming edges is equal to the total flow exiting it from the outgoing edges.

In [33] it is shown that Flow Orientation is NP-complete even if edge labels are $O(\sqrt{n})$, where n is the number of vertices of the graph.

Recall that an edge with label $[k]$ is denoted a $[k]$-edge. You can replace a $[k]$-edge (u, v) by using exclusively $[k/2]$-edges and $[k/4]$-edges as follows.

Construction 1 (refer to Fig. 9b): in addition to u and v introduce vertices x, y, z, $[k/2]$-edges (u, x), (u, y), (z, v), and $[k/4]$-edges (x, z), (y, z), (x, v), (y, v).

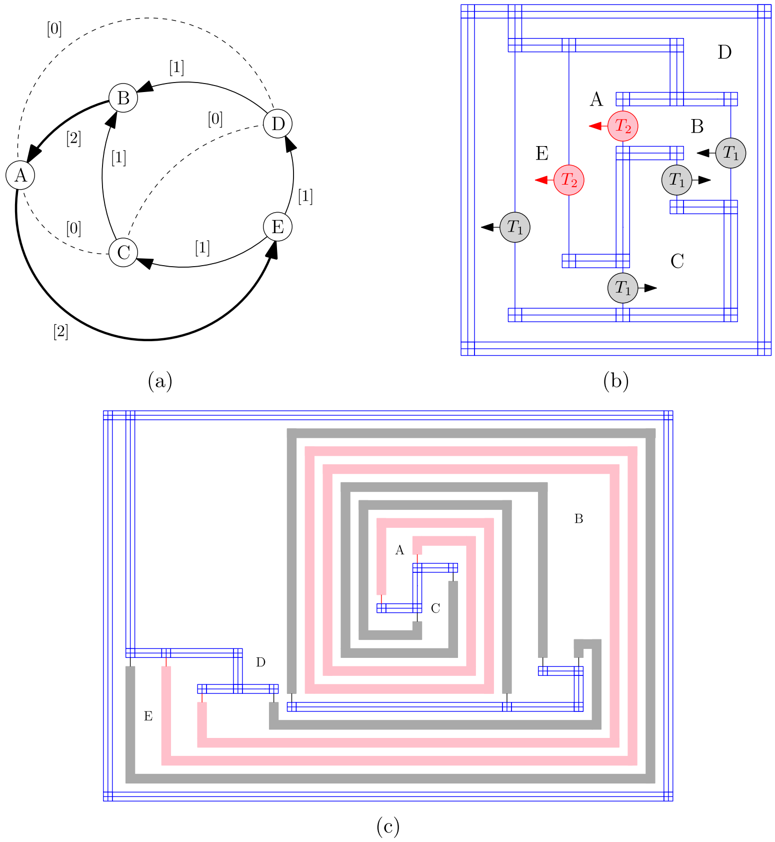


Fig. 8. A positive instance of 1-2-SWITCH-FLOW and the corresponding positive instance of rectilinear-upward planarity testing. (a) An orientation of G^+. (b) The instance of rectilinear-upward planarity testing. (c) A solution of the instance of rectilinear-upward planarity testing.

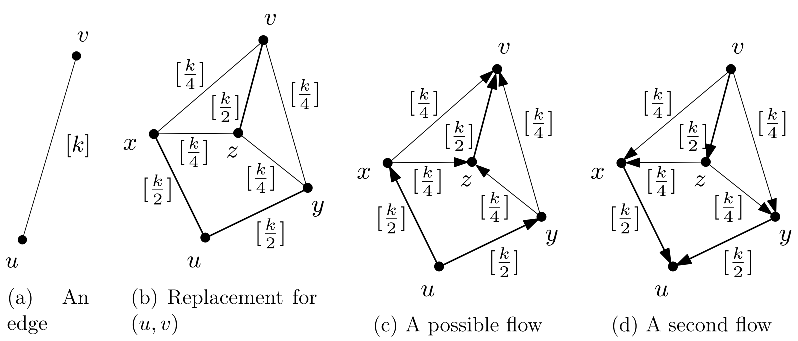


Fig. 9. Replacing an edge with label $[k]$ with edges with labels $[\frac{k}{2}]$ and $[\frac{k}{4}]$.

Observe that the graph of Construction 1 admits a planar embedding with the vertices u and v on the external face.

Lemma 4. *A flow from u to v (or vice versa) in Construction 1 has value k.*

Proof. Suppose that edge (u, x) is traversed by $\frac{k}{2}$ units of flow from u to x (refer to Fig. 9c). By the flow balance at x, edges (x, v) and (x, z) are traversed by $\frac{k}{4}$ units of flow entering v and z, respectively. By the flow balance at z, edge (y, z) is traversed by $\frac{k}{4}$ units of flow exiting y and edge (z, v) is traversed by $\frac{k}{2}$ units of flow entering v. Finally, by the flow balance at y, edge (u, y) is traversed by $\frac{k}{2}$ units of flow entering y and edge (y, v) is traversed by $\frac{k}{4}$ units of flow entering v. With analogous reasoning it can be shown that if edge (u, x) is traversed by $\frac{k}{2}$ units of flow from x to u all the flows through Construction 1 are reversed (refer to Fig. 9c). □

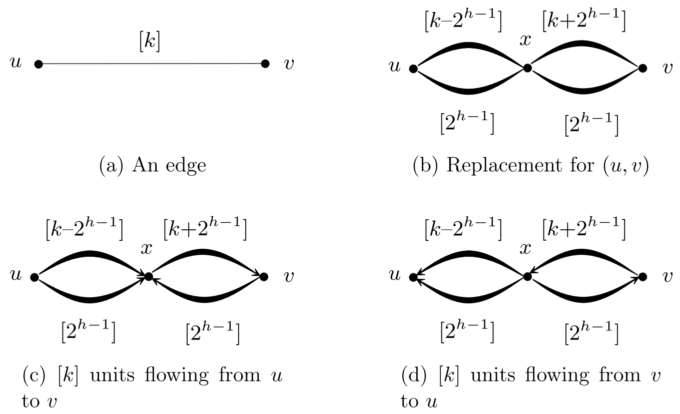


Fig. 10. Replacing an edge with label $[k]$ with subgraphs representing labels $c_1 = k - 2^{h-1}$, $c_2 = k + 2^{h-1}$ and 2^{h-1}.

Based on Lemma 4, we can replace one [4]-edge by using four [1]-edges and three [2]-edges. By iterating the replacement process, we can replace edges of arbitrary high label (provided that it is a power of two).

Lemma 5. *A $[k]$-edge, with $k = 2^h$, can be replaced by an equivalent subgraph, composed exclusively of [1]-edges and [2]-edges, that has a number of edges linear in k.*

Proof. Denote by $e(k)$ the number of [1]-edges plus [2]-edges of the subgraph replacing a $[k]$-edge. By Lemma 4 we are able to replace one $[2^h]$-edge by using $e(2^h) = 3e(2^{h-1}) + 4e(2^{h-2})$ [1]-edges or [2]-edges. Denote by $E_h = e(2^h)$. We have the recursion equation $E_h = 3E_{h-1} + 4E_{h-2}$, with $E_1 = E_2 = 1$, which solves in $E_h = \frac{2}{5}4^h + \frac{3}{5}(-1)^h$ [37]. Since $k = 2^h$, we have that $e(k) = e(2^h) = E_h = \frac{4}{5}k + \frac{3}{5}(-1)^h \le \frac{4}{5}k + \frac{3}{5}$. Therefore, the number of [1]-edges and [2]-edges used to simulate a $[2^h]$-edge is linear in k. $\square$

Now we would like to replace also an edge that has a label that is intermediate between $[2^h]$ and $[2^{h+1}]$. Observe that the gap between such labels $2^{h+1} - 2^h = 2^h$ is also a power of two. Suppose that c_1 and c_2, with $c_1 < c_2$, are two labels (not necessarily powers of two), such that we are able to emulate both a $[c_1]$-edge and a $[c_2]$-edge with suitable subgraphs that only use [1]- and [2]-edges. We show that if $c_2 - c_1$ is a power of two ($c_2 - c_1 = 2^h$), we can emulate a $[k]$-edge with the intermediate value of label $k = \frac{c_1+c_2}{2}$. We introduce the following construction that uses two additional $[2^{h-1}]$-edges.

Construction 2 (refer to Fig. 10b): Add a vertex x. Replace edge (u, v) with a parallel of a $[c_1]$-edge and $[2^{h-1}]$-edge from u to x in series with a parallel of a $[c_2]$-edge and a $[2^{h-1}]$-edge from x to v.

Observe that, assuming that $[c_1]$-edges and $[c_2]$-edges can both be emulated by planar graphs of [1]-edges and [2]-edges such that the starting and ending vertices are on the external face, Construction 2 guarantees that also a $[\frac{c_1+c_2}{2}]$-edge can be emulated by a planar graph of [1]-edges and [2]-edges such that the starting and ending vertices are on the external face.

Lemma 6. *A flow from u to v (or vice versa) in Construction 2 has value $\frac{c_1+c_2}{2}$.*

Proof. The first parallel from u to x of Fig. 10b allows for a $c_1 - 2^{h-1}$ flow or a $c_1 + 2^{h-1} = k$ flow flowing either from u to x or from x to u. The second parallel from x to v allows for a $c_2 - 2^{h-1} = k$ or a $c_2 + 2^{h-1}$ flow flowing either from x to v or from v to x. The balance of flow at vertex x imposes that $c_1 + 2^{h-1} = c_2 - 2^{h-1} = k$ units of flow are flowing either from u to v or from v to u. $\square$

Now, we use Lemma 6 to build a replacement for an edge that has a generic label $[k]$ with $2^h < k < 2^{h+1}$.

Lemma 7. *A $[k]$-edge, with $2^h < k < 2^{h+1}$ can be replaced by an equivalent subgraph composed exclusively of [1]-edges and [2]-edges whose number of [1]-edges plus [2]-edges is $e(k) = O(k \log(k))$.*

Proof. Our strategy is that of iteratively applying Lemma 6 as if we were performing a binary search for k in the interval $[2^h, 2^{h+1}]$, simulating at each step the intermediate value between the extremes of the interval and comparing that value with k. We start with the interval $[c_1, c_2]$, with $c_1 = 2^h$ and $c_2 = 2^{h+1}$. We apply Lemma 6 and obtain a replacement for an edge with label $[\frac{c_1+c_2}{2}]$ with a number $e(\frac{c_1+c_2}{2}) = e(2^h) + e(2^{h+1}) + 2e(2^h) < 4e(2^{h+1})$ of [1]-edges or [2]-edges. If $\frac{c_1+c_2}{2} = k$,

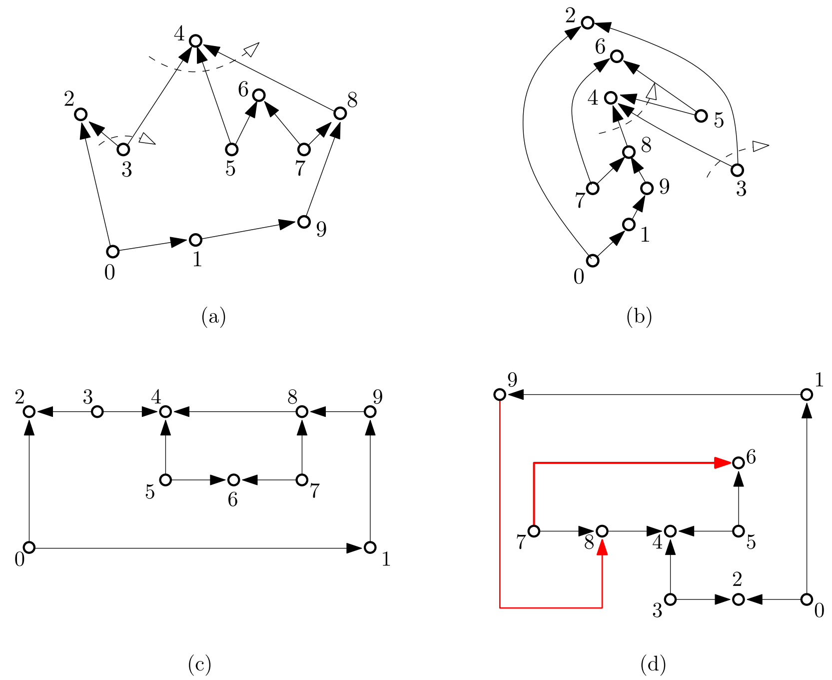


Fig. 11. (a)-(b) Two upward planar embeddings of the same digraph, within the same planar embedding; for instance, the left-to-right ordering of the edges at vertices 3 and 4 differ in the two upward planar embeddings. (c) An RU representation for the upward planar embedding in (a). (d) The upward embedding in (b) does not have any RU representation (some edges need bends or downward segments).

we are done. If $\frac{c_1+c_2}{2} > k$ we iterate replacing c_2 with $\frac{c_1+c_2}{2}$. Otherwise ($\frac{c_1+c_2}{2} < k$), we iterate replacing c_1 with $\frac{c_1+c_2}{2}$. At each iteration we construct a replacement for a $[\frac{c_1+c_2}{2}]$-edge using at most $2e(2^{h+1})$ [1]-edges or [2]-edges in addition to $e(c_1) + e(c_2)$ [1]-edges or [2]-edges that we already accounted for. Since the number of iterations in the worst case is h, we have that the final replacement for a $[k]$-edge uses $4e(2^{h+1}) + 2he(2^{h+1})$ [1]-edges or [2]-edges. Since $2^{h+1} = O(k)$ and $h = O(\log(k))$, we have that $e(k) = O(k \log(k))$. □

Theorem 2. 1-2-Switch-Flow *is NP-complete.*

Proof. First observe that 1-2-Switch-Flow is in NP because, given an instance G' of 1-2-Switch-Flow, one could non-deterministically guess the flow along each edge and then check in polynomial time if each vertex is balanced, i.e., if the guessed flow is a solution of the 1-2-Switch-Flow instance. Regarding the hardness: consider an n-vertex planar graph G that is an instance of Flow Orientation where the labels of the edges of G are bounded by $\sqrt{n}$. Apply Lemma 5 or 7 to each edge of G replacing it with at most $O(\sqrt{n}\log(\sqrt{n})) = O(\sqrt{n}\log(n))$ [1]-edges or [2]-edges. You obtain an instance G' of 1-2-Switch-Flow whose size $O(n\sqrt{n}\log(n))$ is polynomial in the size of G. The equivalence of the atomic replacements proved in Lemmas 4 and 6 implies that G' is a yes instance of 1-2-Switch-Flow if and only if G is a yes instance of Flow Orientation. □

5. Testing upward plane digraphs in linear time

If we have in input a rectilinear planar representation H of a digraph G, testing whether H is also an RU representation for one of the four possible orientations of H is a trivial problem. On the contrary, given an upward plane digraph G, testing whether it admits an RU representation is a relevant problem. In this section we provide a linear-time algorithm for this problem. We recall that an early paper in the graph drawing literature [38] claims the result of this section. Unfortunately, that paper only gives a sketch about the algorithm to test RU planarity without giving sufficient details and simultaneously referring to a more restrictive model [39].

Our algorithm takes as input an upward plane digraph $G = (V, E)$ and tests whether G admits an RU representation that preserves its upward planar embedding. It is important to observe that, for any given planar embedding of G, there might be exponentially many upward planar embeddings within that planar embedding. An RU representation might exists for some upward planar embeddings, while for other upward planar embeddings a solution may not exist. See for example Fig. 11.

The idea of our approach is to compute a *candidate set of labels* $\lambda(e)$ for each edge e of G, which corresponds to the possible directions for e in some RU representation, if one exists. The labels in $\lambda(e)$ are taken in the set $\{L, U, R\}$, where U,

L, and R denote the upward, the leftward, and the rightward directions, respectively. Once the sets of candidate labels for the edges of G have been computed, we exploit a 2-SAT formulation to decide whether it is possible to select exactly one candidate label for each edge, such that the whole set of edge directions defines a valid RU representation. We will show that the candidate set of labels $\lambda(e)$ for each edge $e = (u, v)$ can be easily computed from two initial sets $\lambda_{\text{in}}(e)$ and $\lambda_{\text{out}}(e)$ of local labels for e, which encode the sides (east, west, north) by which e can leave and enter its end-vertices according to the upward planar embedding of G.

Following the aforementioned strategy, our algorithm consists of three phases, summarized hereunder and then described in more detail.

- Phase 1: For each edge $e = (u, v)$, we assign two initial sets of labels $\lambda_{\text{out}}(e)$ and $\lambda_{\text{in}}(e)$ to e, each label taken in the set $\{E, W, N\}$ (East, West, or North). The set $\lambda_{\text{out}}(e)$ encodes the sides by which e can leave u; the set $\lambda_{\text{in}}(e)$ encodes the sides by which e can enter v.
- Phase 2: We use the initial label assignment of Phase 1 to compute the candidate edge labeling λ of G. If $\lambda(e)$ is empty, for some edge e, the input graph does not admit any RU representation.
- Phase 3: By means of the candidate set of labels of the edges of G, RU planarity is modeled as a 2-SAT formula ϕ, which is then solved in linear time [40,41].

Details for Phase 1. We describe how to define the sets $\lambda_{\text{out}}(\cdot)$ and $\lambda_{\text{in}}(\cdot)$ for every edge outgoing or incoming a vertex w of G. (i) If w has three outgoing (resp. incoming) edges e_1, e_2, e_3, in this left-to-right order in the upward planar embedding of G, then the sides from which these edges are incident to w can be uniquely fixed. Namely, $\lambda_{\text{out}}(e_1) = \{W\}$, $\lambda_{\text{out}}(e_2) = \{N\}$, $\lambda_{\text{out}}(e_3) = \{E\}$ (resp. $\lambda_{\text{in}}(e_1) = \{W\}$, $\lambda_{\text{in}}(e_2) = \{S\}$, $\lambda_{\text{in}}(e_3) = \{E\}$). ($ii$) If w has two outgoing (resp. incoming) edges e_1 and e_2, in this left-to-right order, we set $\lambda_{\text{out}}(e_1) = \{W, N\}$, $\lambda_{\text{out}}(e_2) = \{N, E\}$ (resp. $\lambda_{\text{in}}(e_1) = \{W, S\}$, $\lambda_{\text{in}}(e_2) = \{S, E\}$). ($iii$) If w has one outgoing edge e_1 we set $\lambda_{\text{out}}(e_1) = \{W, N, E\}$ in all cases except when w has three incoming edges, in which case $\lambda_{\text{out}}(e_1) = \{N\}$. ($iv$) If w has one incoming edge e_1 we set $\lambda_{\text{in}}(e_1) = \{W, S, E\}$ in all cases except when w has three outgoing edges, in which case $\lambda_{\text{in}}(e_1) = \{S\}$.

Details for Phase 2. For each edge e, given the label sets $\lambda_{\text{out}}(e)$ and $\lambda_{\text{in}}(e)$ for e, we first initialize $\lambda(e)$ as the empty set. If $S \in \lambda_{\text{in}}(e)$ and $N \in \lambda_{\text{out}}(e)$, we add label U to $\lambda(e)$. If $E \in \lambda_{\text{out}}(e)$ and $W \in \lambda_{\text{in}}(e)$, we add label R to $\lambda(e)$. If $W \in \lambda_{\text{out}}(e)$ and $E \in \lambda_{\text{in}}(e)$, we add label L to $\lambda(e)$. We say that λ is a *good labeling* if it exists an RU representation H of G such that each edge $e \in H$ has a direction that corresponds to one of the labels of $\lambda(e)$; if so, H is said to be *compatible* with λ. Note that the labeling λ constructed as described above is such that, for each edge $e = (u, v)$, $|\lambda(e)| \leq 3$ and the label U always belongs to $\lambda(e)$ when $|\lambda(e)| \geq 2$. Also, if $|\lambda(e)| = 3$ then $\lambda(e) = \{L, U, R\}$, and e is the only outgoing edge of u and the only incoming edge of v. Consider another labeling λ', derived from λ as follows: If $|\lambda(e)| \leq 2$, let $\lambda'(e) = \lambda(e)$; if $|\lambda(e)| = 3$, let $\lambda'(e) = \{U\}$. Clearly, λ' is constructed in linear time from λ and $|\lambda'(e)| \leq 2$, for every edge e of the graph. We call λ' the *reduction of* λ. The following lemma is crucial for our 2-SAT model.

Lemma 8. *λ is a good labeling if and only if its reduction λ' is a good labeling.*

Proof. Clearly, if λ' is a good labeling then λ is, because $\lambda(e)$ is a superset of $\lambda'(e)$. Suppose, vice versa, that λ is a good labeling. We prove that λ' is a good labeling by induction on the number k of edges e for which $|\lambda(e)| = 3$. If $k = 0$, λ and λ' coincides, and the statement is obvious. Suppose that the statement is true for any $k \geq 1$, and let e be any edge for which $|\lambda(e)| = 3$. Let λ'' be the labeling obtained from λ' by setting $\lambda''(e) = \lambda(e) = \{L, U, R\}$. By the inductive hypothesis, λ'' is a good labeling. Consider an RU representation H of G compatible with λ and let d be the direction of e in H. If d is the upward direction, then H is also compatible with λ', because $\lambda'(e) = \{U\}$. Otherwise (i.e., d is either the rightward or the leftward direction) there is neither an edge of H that leaves u from North nor an edge of H that enters v from South (because e is the only outgoing edge of u and the only incoming edge of v). Hence, we can derive from H another RU representation H' such that e points upward while all other edges of H' have the same direction as in H. The representation H' is now compatible with λ', which implies that λ' is a good labeling. □

Details for Phase 3. By Lemma 8, we can always assume that the labeling λ determined in the previous phase is such that $\lambda(e)$ contains either one or two labels, for each edge e of G. Indeed, if this is not the case, we can restrict to consider its reduction λ', obtained from λ in linear time. Let w be a vertex of G and let e_1 and e_2 be two edges of G that are either both outgoing w or both incoming w. Two labels $X \in \lambda(e_1)$ and $Y \in \lambda(e_2)$ are *conflicting* if $X = Y$. Indeed, there cannot exist an RU representation of G such that the directions of e_1 and e_2 coincide. Let e_1 be an edge outgoing w and let e_2 be an edge incoming w. Two labels $X \in \lambda(e_1)$ and $Y \in \lambda(e_2)$ are *conflicting* if X and Y represent *opposite directions* (i.e., $X = L$ and $Y = R$ or $X = R$ and $Y = L$). This phase aims to assign a single label to each edge so that there are no conflicting labels. Such an assignment (if any) is a *non-conflicting label assignment within* λ. We use the notation $\mathcal{L}(\lambda)$ to denote any non-conflicting assignment within λ. The next lemma establishes an equivalence between non-conflicting label assignments and RU representations of G compatible with λ.

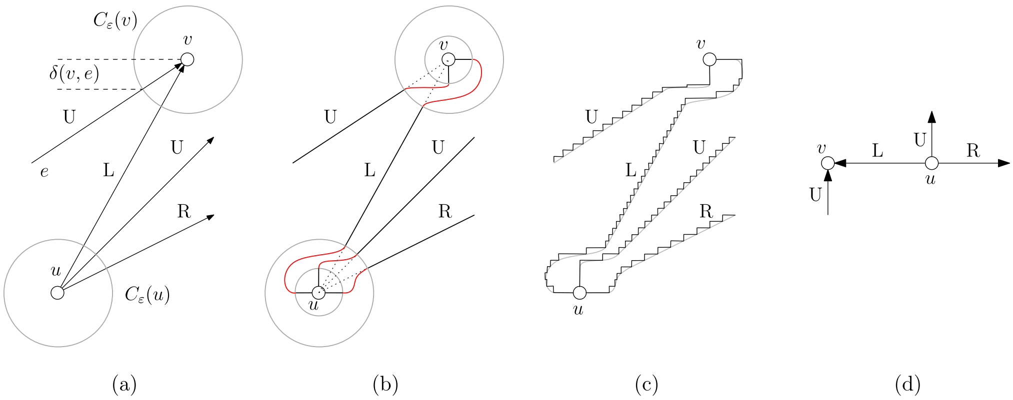


Fig. 12. An illustration for the proof of Lemma 9.

Lemma 9. *Let λ be a candidate edge labeling of G. There exists an RU representation H that is compatible with λ if and only if there exists a non-conflicting label assignment within λ. The edge directions defined by H correspond to those defined by the label assignment, and H preserves the planar embedding of G.*

Proof. If there exists an RU representation H that is compatible with λ, then choosing for each edge e the label of $\lambda(e)$ that corresponds to the direction of e in H immediately yields a non-conflicting label assignment within λ.

Suppose vice versa that there exists a non-conflicting label assignment $\mathcal{L}(\lambda)$. We show that from $\mathcal{L}(\lambda)$ we can derive an RU representation H of G that is compatible with λ. Since by hypothesis G is upward planar and comes with an upward planar embedding, there exists a straight-line upward planar drawing Γ' of G that preserves its upward planar embedding [42]. We can construct from Γ' an orthogonal-upward drawing Γ'' of G such that for each edge $e = (u, v)$: (*i*) the directions of the segments of e that are incident to u and v are coherent with the label of e in $\mathcal{L}(\lambda)$; and (*ii*) moving from u to v along e, the number of right bends equals the number of left bends.

To construct Γ'', proceed as follows. Let $\varepsilon > 0$ be a length such that the circular area of radius ε around each vertex v does not intersect any other vertex and any other edge that is not incident to v in Γ'. For each vertex v of Γ', draw a circle $C_\varepsilon(v)$, centered at v, of radius ε (see Fig. 12a). Let e be an edge incident to v. Denote by $\delta(v, e)$ the smallest vertical distance between v and the intersection of e with $C_\epsilon(v)$ (see Fig. 12a). Let δ be the minimum of all $\delta(v, e)$. Draw a circle $C_\delta(v)$ of radius δ around each vertex v. Now construct a drawing of G such that each edge $e = (u, v)$ is non-decreasing with respect to the y-coordinate, leaves the u vertex and enters vertex v with a straight segment of length δ directed as prescribed by λ. This is possible because $\delta = \min_{v,e}\{\delta(v, e)\}$ (see Fig. 12b). To finally obtain Γ'', we replace each edge e by a sequence of horizontal and vertical segments that follows the drawing of e at a distance that is small enough to guarantee that it does not intersect any other edge or vertex of the drawing (see Fig. 12a). Since the sequence of horizontal and vertical segments of each edge e starts and ends with a segment that goes in the same direction and is non-decreasing, the numbers of right and left turns are the same.

To construct the final RU representation H, consider the orthogonal representation H'' of Γ''. Since each edge of H'' has the same number of left and right turns, by [7] there exists an orthogonal representation H of G without bends (i.e., a rectilinear representation of G) such that H has the same embedding as H'' and such that each edge in H is incident to its end-vertices from the same side as in H'' (see Fig. 12d). $\square$

Given a non-conflicting label assignment $\mathcal{L}(\lambda)$ of G, an RU representation H of G compatible with λ and whose edge directions correspond to the edge labels of $\mathcal{L}(\lambda)$, can be easily constructed in linear time. Namely, since H preserves the planar embedding of G, for each vertex w of H the angles at w can be easily determined by the label assignment $\mathcal{L}(\lambda)$ for the edges incident to w. Also, H is oriented in such a way that the directions of the edges are coherent with λ. We now give the main result of this section.

Theorem 3. *Let G be an n-vertex upward plane digraph. There exists an $O(n)$-time algorithm that tests whether G admits an RU representation, and that computes one in the positive case.*

Proof. The algorithm first applies Phase 1 and Phase 2 described above. Denote by λ the candidate edge labeling of G at the end of Phase 2. By Lemma 8, we can assume that, for each edge e of G, $\lambda(e)$ contains at most two labels. Based on Lemma 9, deciding whether G admits an embedding-preserving RU representation is equivalent to deciding whether G admits a non-conflicting labeling $\mathcal{L}(\lambda)$. We model this problem as a 2-SAT problem, which is defined as follows.

For each edge e and for each label $X \in \lambda(e)$, define a Boolean variable b_e^X; this variable will be set to `True` if we select label X for edge e, and it will be set to `False` otherwise. We define a formula $cl(e)$ for every edge e and a formula $cl(v)$

for every vertex v that has at least one incident edge e with $|\lambda(e)| = 2$. Our 2-SAT formula Φ is the conjunction of all the formulas defined for the edges and for the vertices of G.

For each edge e of G we define $cl(e)$ as the conjunction of two clauses. Namely, if $\lambda(e) = \{X, Y\}$ we have $cl(e) = (b_e^X \vee b_e^Y) \wedge (\neg b_e^X \vee \neg b_e^Y)$. This ensures that in order to satisfy Φ we have to select exactly one of the two labels X and Y. If $\lambda(e) = \{X\}$, we want to have $b_e^X = \texttt{True}$, hence we introduce a dummy binary variable b and we let $cl(e) = (b_e^X \vee b) \wedge (b_e^X \vee \neg b)$.

For each vertex v incident to at least one edge with two candidate labels, we define $cl(v)$ based on two cases: (i) v is a source or a sink; (ii) v is neither a source nor a sink.

Case (i). If v is a source (resp. a sink), let e_1 and e_2 be the two outgoing (resp. incoming) edges from v, respectively. We set:

$$cl(v) = \neg b_{e_1}^U \vee \neg b_{e_2}^U$$

Case (ii). We have four subcases: (a) $\deg(v) = 2$; (b) $\deg(v) = 3$ and v has two incoming edges; (c) $\deg(v) = 3$ and v has two outgoing edges; (d) $\deg(v) = 4$ and v has two incoming and two outgoing edges.

(a) Let e_1 and e_2 be the incoming edge and the outgoing edge of v, respectively. Suppose that $X \in \lambda(e_1)$ and $Y \in \lambda(e_2)$, where $X, Y \in \{R, L\}$. If $X = Y$, we do not add any clause associated with v, because any two labels for e_1 and e_2 in $\lambda(e_1)$ and in $\lambda(e_2)$ are non-conflicting. If $X \neq Y$, we set:

$$cl(v) = \neg b_{e_1}^X \vee \neg b_{e_2}^Y.$$

(b) Let e_1 and e_2 be the two incoming edges of v, in this left-to-right order, and let e_3 be the outgoing edge of v. Suppose first that $|\lambda(e_1)| = |\lambda(e_2)| = |\lambda(e_3)| = 2$. Then we have $\lambda(e_1) = \{U, R\}$, $\lambda(e_2) = \{L, U\}$, and either (1) $\lambda(e_3) = \{L, U\}$ or (2) $\lambda(e_3) = \{U, R\}$. We define $cl(v)$ as follows, depending on the two sub-cases:

$$(1)\ cl(v) = (\neg b_{e_1}^U \vee \neg b_{e_2}^U) \wedge (\neg b_{e_1}^R \vee \neg b_{e_3}^L)$$
$$(2)\ cl(v) = (\neg b_{e_1}^U \vee \neg b_{e_2}^U) \wedge (\neg b_{e_2}^L \vee \neg b_{e_3}^R)$$

If $|\lambda(e_i)| = 1$, for some $i \in \{1, 2, 3\}$, we must add fewer constraints to the formula for avoiding conflicting labels, thus $cl(v)$ is simpler and is the conjunction of a subset of the clauses in (1) and (2).

(c) Symmetric to (b).

(d) Let e_1 and e_2 be the two outgoing edges of v, and let e_3 and e_4 be the two incoming edges of v, in this left-to-right order. As above, in the worst case we have $\lambda(e_1) = \{L, U\}$, $\lambda(e_2) = \{U, R\}$, $\lambda(e_3) = \{L, U\}$, $\lambda(e_4) = \{U, R\}$, and we define $cl(v)$ as follows:

$$cl(v) = (\neg b_{e_1}^U \vee \neg b_{e_2}^U) \wedge (\neg b_{e_2}^R \vee \neg b_{e_3}^L) \wedge (\neg b_{e_3}^U \vee \neg b_{e_4}^U) \wedge (\neg b_{e_1}^L \vee \neg b_{e_4}^R)$$

If $|\lambda(e_i)| = 1$, for some $i \in \{1, 2, 3, 4\}$, we must add fewer constraints for avoiding conflicting labels, thus $cl(v)$ is simpler and is the conjunction of a subset of the clauses above.

The size of the 2-SAT formula constructed with the procedure above is $O(n)$, as the number of (binary) clauses per element (vertex or edge) is bounded by a constant. The time complexity follows by the fact that 2-SAT is linear-time solvable, for instance via an implication-graph implementation or via a backtracking algorithm [40,41]. $\square$

6. Testing in the variable embedding setting

In this section we deal with biconnected planar digraphs whose embedding is not fixed. In Section 6.1 we define the notion of rectilinear-upward spirality. In Section 6.2 we describe a polynomial-time testing algorithm for biconnected digraphs whose underlying undiracted graph is series-parallel. In Section 6.3 we consider the general case, and give an FPL algorithm parameterized by the number of sources and sinks in the digraph.

6.1. Rectilinear-upward spirality

We introduce the new concept of rectilinear-upward spirality, which specializes the notion of orthogonal spirality defined in [8]. While the orthogonal spirality is a measure of how much a given subgraph of an undirected graph G is rolled-up in an orthogonal representation of G, our notion of spirality is for directed graphs and incorporates additional information about the sides to which edges are incident to the poles of the triconnected components of G.

SPQR-trees. Let G be a biconnected planar (di)graph. The *SPQR-tree* T of G, introduced in [1], represents the decomposition of G into its triconnected components [43]; see Fig. 13 and Fig. 14. Each triconnected component corresponds to a non-leaf node ν of T; the triconnected component itself is the *skeleton* of ν and is denoted as $\text{skel}(\nu)$. Node ν can be: (i) an *S-node* (series composition), if $\text{skel}(\nu)$ is a simple cycle of length at least three; (ii) a *P-node* (parallel composition), if $\text{skel}(\nu)$ is a bundle of at least three parallel edges; (iii) an *R-node* (rigid composition), if $\text{skel}(\nu)$ is a triconnected graph. A degree-1

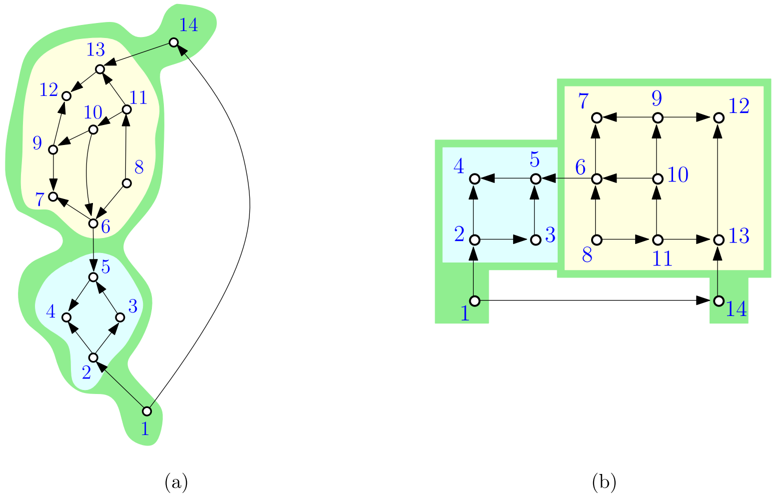


Fig. 13. (a) A digraph G with three highlighted components (an S-, an R- and a P-component); (b) an RU representation of G.

node of T is a *Q-node* and represents a single edge of G. A *real edge* (resp. *virtual edge*) in $\mathrm{skel}(\nu)$ corresponds to a Q-node (resp., to an S-, P-, or R-node) adjacent to ν in T. Neither two S- nor two P-nodes are adjacent in T. The SPQR-tree of a biconnected graph can be computed in linear time [1,44]. Let e be a designated edge of G, called the *reference edge* of G, let ρ be the Q-node of T corresponding to e, and let T be rooted at ρ. For any P-, S-, or R-node ν of T distinct from the root child, $\mathrm{skel}(\nu)$ has a virtual edge, called *reference edge* of $\mathrm{skel}(\nu)$ and of ν, associated with a virtual edge in the skeleton of its parent. The reference edge of the root child of T is the edge corresponding to ρ. The end-vertices of the reference edge of a node ν are the *poles* of ν (and of $\mathrm{skel}(\nu)$). The rooted tree T describes all planar embeddings of G with its reference edge on the external face; they are obtained by combining the different planar embeddings of the skeletons of P- and R-nodes with their reference edges on the external face. Namely, for a P- or R-node ν, denote by $\mathrm{skel}^-(\nu)$ the skeleton of ν without its reference edge. If ν is a P-node, the embeddings of $\mathrm{skel}(\nu)$ are the different permutations of the edges of $\mathrm{skel}^-(\nu)$; If ν is an R-node, $\mathrm{skel}(\nu)$ has two possible embeddings, obtained by flipping $\mathrm{skel}^-(\nu)$ at its poles. For every node $\nu \neq \rho$, the *pertinent graph* G_ν *of* ν is the subgraph of G whose edges correspond to the Q-nodes in the subtree of T rooted at ν. We also say that G_ν is a *component of* G. The pertinent graph G_ρ of the root ρ coincides with the reference edge of G. If H is a rectilinear representation or an RU rectilinear representation of G, its restriction H_ν to G_ν is a *rectilinear component* or an *RU rectilinear component* of H. As in [20,8,21,27,45,46,33], we implicitly assume to work with a *normalized* SPQR-tree, in which every S-node has exactly two children. Indeed, every SPQR-tree can be transformed in linear time into a normalized SPQR-tree by recursively splitting an S-node with more than two children into multiple S-nodes with two children. If G has n vertices, a normalized SPQR-tree of G still has $O(n)$ nodes. Also, observe that the sum of the sizes of all skeletons for the nodes of an SPQR-tree (and of a normalized SPQR-tree) is $O(n)$ [47].

RU-spirality. Let G be a biconnected planar digraph and consider an SPQR-tree T of G rooted at a Q-node ρ, corresponding to a reference edge (s,t). Assume for convenience that the vertices of G are labeled with an st-numbering [48] of G (see, e.g., Fig. 13a). Let H be an orthogonal representation of G with the reference edge $G_\rho = (s,t)$ in the external face, let H_ν be a component of H (i.e., the restriction of H to G_ν), and let $\{u, v\}$ be the poles of ν, where u precedes v in the st-numbering. We say that u and v are the *first-pole* and the *second-pole* of ν, respectively. Note that we are not assuming any relationship between the st-numbering and the orientation of the edges of G. For each pole $w \in \{u, v\}$, let $\mathrm{intdeg}_\nu(w)$ and $\mathrm{extdeg}_\nu(w)$ be the degree of w inside and outside H_ν, respectively. We define two (possibly coincident) *alias vertices* of w, denoted by w' and w'', as follows: (i) if $\mathrm{intdeg}_\nu(w) = 1$, then $w' = w'' = w$; (ii) if $\mathrm{intdeg}_\nu(w) = \mathrm{extdeg}_\nu(w) = 2$, then w' and w'' are dummy vertices, each splitting one of the two distinct edge segments incident to w outside H_ν; (iii) if $\mathrm{intdeg}_\nu(w) > 1$ and $\mathrm{extdeg}_\nu(w) = 1$, then $w' = w''$ is a dummy vertex that splits the edge segment incident to w outside H_ν.

Let A^w be the set of distinct alias vertices of a pole w. Let P^{uv} be any simple undirected path from u to v inside H_ν and let $u' \in A^u$ and $v' \in A^v$ be two alias vertices of u and of v, respectively. The path $S^{u'v'}$ obtained concatenating (u', u), P^{uv}, and (v, v') is a *spine* of H_ν. Denote by $n(S^{u'v'})$ the number of right turns minus the number of left turns encountered along $S^{u'v'}$ moving from u' to v'. The *rectilinear spirality* $\sigma(H_\nu)$ of H_ν is either an integer or a semi-integer number, defined based on the following cases: (i) If $A^u = \{u'\}$ and $A^v = \{v'\}$ then $\sigma(H_\nu) = n(S^{u'v'})$. (ii) If $A^u = \{u'\}$ and $A^v = \{v', v''\}$ then $\sigma(H_\nu) = \frac{n(S^{u'v'}) + n(S^{u'v''})}{2}$. (iii) If $A^u = \{u', u''\}$ and $A^v = \{v'\}$ then $\sigma(H_\nu) = \frac{n(S^{u'v'}) + n(S^{u''v'})}{2}$. (iv) If $A^u = \{u', u''\}$ and $A^v = \{v', v''\}$ assume, without loss of generality, that (u, u') succeeds (u, u'') clockwise around u and that (v, v') precedes (v, v'') clockwise around v; then $\sigma(H_\nu) = \frac{n(S^{u'v'}) + n(S^{u''v''})}{2}$.

For brevity, in the following we often denote by σ_ν the rectilinear spirality of an RU representation of G_ν. Let $\{S, N, W, E\}$ denote the set of the four possible sides (North, South, East, West) by which an edge can be incident to a vertex in an RU representation. The *rectilinear-upward spirality* (*RU-spirality* for short) of H_ν, denoted by $\tau(H_\nu)$ (or simply by τ_ν), is a tuple

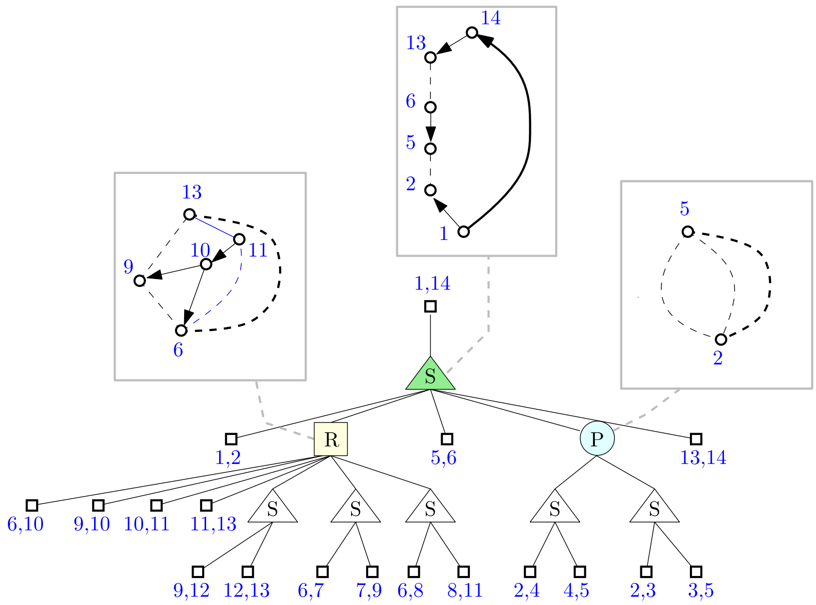


Fig. 14. The SPQR-tree of the graph in Fig. 13, with reference edge $(1, 14)$; the skeletons of the three highlighted components are shown, where dashed edges are virtual and the reference edge is thicker. Q-nodes are labeled with the end-vertices of their corresponding edges.

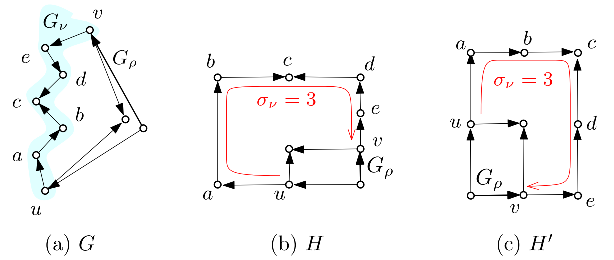


Fig. 15. Illustration of the concept of RU-spirality. The two representations in (b) and (c) have the same value of σ_ν but different RU spiralities.

$\langle \sigma_\nu, \varphi_u, \varphi_v \rangle$, where σ_ν is the rectilinear spirality of H_ν and where $\varphi_w = (S_w, N_w, W_w, E_w)$ specifies the arrangement of the internal and external edges of H_ν incident to a pole $w \in \{u, v\}$, with respect to the four sides S (South), N (North), W (West), and E (East). Precisely, for each $D \in \{S, N, W, E\}$ and $w \in \{u, v\}$, we have $D_w \in \{\text{free}, \text{int}, \text{ext}\}$ in such a way that: $D_w = \text{free}$ if no edge is incident to w from side D; $D_w = \text{int}$ if there is an edge of H_ν (i.e., an edge internal to H_ν) incident to w from side D; $D_w = \text{ext}$ if there is an edge of H not in H_ν (i.e., an edge external to H_ν) incident to w from side D. We call σ_ν the *rectilinear spirality* of τ_ν; φ_u and φ_v the *pole side specifications* of τ_ν. Fig. 15 shows an illustration of the concept of RU-spirality. In Fig. 15a there is a digraph G with a highlighted S-component G_ν with first-pole u and second-pole v. Fig. 15b and Fig. 15c show two different RU representations H and H' of G. In H we have $\tau_\nu = \langle 3, \varphi_u = (\text{free}, \text{ext}, \text{int}, \text{ext}), \varphi_v = (\text{ext}, \text{int}, \text{ext}, \text{free})\rangle$ and in H' we have $\tau_\nu = \langle 3, \varphi_u = (\text{ext}, \text{int}, \text{free}, \text{ext}), \varphi_v = (\text{free}, \text{ext}, \text{ext}, \text{int})\rangle$.

Note that, denoted by G' the subgraph of G consisting of G_ν plus the external edges incident to the poles of ν, the RU-spirality for an RU representation H_ν of G_ν can also be defined referring to an RU representation of G' that contains H_ν, rather than to an RU representation of G. If we are able to construct an RU representation H' of G' such that its restriction H_ν to G_ν has RU-spirality τ_ν, then we say that G_ν (or simply ν) *admits* RU-spirality τ_ν. Observe that, even if H_ν admits RU-spirality τ_ν, it might not exist an RU representation of G whose restriction to G_ν has spirality τ_ν.

Substituting components with the same RU-spirality. We extend the results in [8,45] to show that components with the same RU-spirality are "interchangeable" (see Fig. 16 for an illustration).

Let H be an RU representation of a planar digraph G with a given reference edge on the external face. Let G' be a planar digraph obtained from G by replacing a component G_ν of G with a subgraph G'_ν such that: (*i*) G'_ν has the same poles as G_ν; (*ii*) for each pole w of G_ν, the subgraph G'_ν has the same number of edges entering w as G_ν and the same number of edges leaving w as G_ν; (*iii*) G' admits an RU representation H' such that $\tau(H_\nu) = \tau(H'_\nu)$, where H_ν and H'_ν are the restrictions of H to G_ν and of H' to G'_ν, respectively. The operation $\text{Sub}(H_\nu, H'_\nu)$ of *substituting* H_ν with H'_ν in H defines a new plane digraph H'' with an angle labeling such that the restriction of H'' to G_ν coincides with H'_ν, while the restriction of H'' to $G \setminus G_\nu$ stays as in H. More formally, let u and v be the first-pole and second-pole of ν, respectively. The external boundary of H_ν contains a left path p_l and a right path p_r, such that p_l (resp. p_r) goes from u to v traversing the external

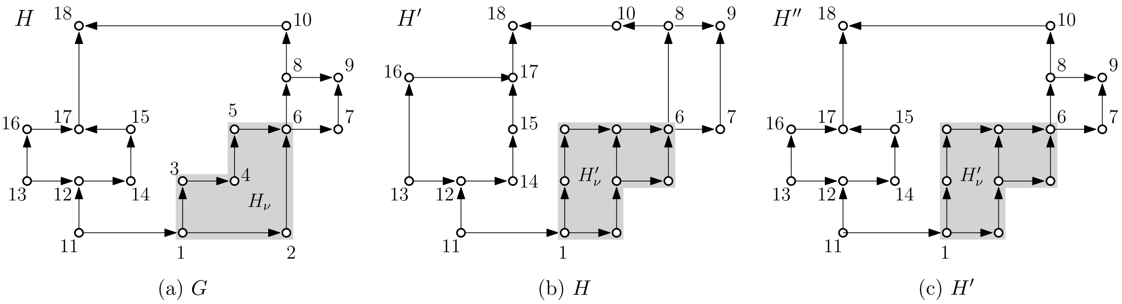


Fig. 16. Illustration of the concept of substitution. The RU representation H'' in (c) is obtained by substituting H_ν with H'_ν in H. The first-pole of ν is vertex 1 and the second-pole of ν is vertex 6. We have $\tau(H_\nu) = \tau(H'_\nu) = \langle -\frac{1}{2}, (\text{free}, \text{int}, \text{ext}, \text{int}), (\text{int}, \text{ext}, \text{int}, \text{ext})\rangle$.

boundary of H_ν clockwise (resp. counterclockwise). Let f_l and f_r be the faces of H outside H_ν and incident to p_l and p_r, respectively. With respect to H'_ν and H', define p'_l, p'_r, f'_l, f'_r analogously. Since $\tau(H_\nu) = \tau(H'_\nu)$, the circular sequence of angles at each pole $w \in \{u, v\}$ is the same in H and in H', namely the angles at w internal and external to G_ν are the same in H and H'. The digraph H'' is defined as follows:

- H'' has the same set of vertices and edges as G.
- The planar embedding of H'' is such that: all the faces of H outside H_ν and distinct from f_l and f_r, as well as all faces of H'_ν, are also faces of H''. Further, H'' has two faces f''_l and f''_r obtained by replacing p_l with p'_l and p_r with p'_r in the boundary of f_l and of f_r, respectively.
- The angle labeling of H'' is such that: (i) all the angles at the vertices of G not belonging to G_ν are those in H; (ii) all the angles at the vertices of G_ν distinct from u and v are those in H'_ν; (iii) for each pole $w \in \{u, v\}$, the internal and external angles at w are defined as in H or, equivalently, as in H' (they are the same because $\tau(H_\nu) = \tau(H'_\nu)$).

The following result proves that H'' is an RU representation.

Theorem 4. *Let G be a biconnected planar digraph, T be an SPQR-tree of G with respect to a given reference edge e, and ν be a non-root node of T. Let G' be a planar digraph obtained from G by replacing G_ν with another component having the same poles u and v as G_ν and, for each pole $w \in \{u, v\}$, the same number of edges entering w and leaving w as in G_ν. Suppose that H is an RU representation of G with e on the external face and suppose that H' is an RU representation of G' with e on the external face. Also, let H_ν and H'_ν be the restrictions of H and of H' to G_ν and to G'_ν, respectively. If $\tau(H_\nu) = \tau(H'_\nu)$ then the graph H'' defined by* $\text{Sub}(H_\nu, H'_\nu)$ *is an RU representation of G.*

Proof. The fact that the planar embedding and the labeling of H'' describe a rectilinear planar representation of G is proved in [8,45], as a consequence of the fact that H_ν and H'_ν have the same rectilinear spirality. We now orient H'' in such a way that, for an arbitrarily chosen edge $e = (x, y)$ of H'_ν, the vertices x and y have the same relative positions as in H'_ν. This implies that for each edge e' of G_ν, the relative position of the end-vertices of e' in H'' remains the same as in H'. Also, for each side $\{S, N, W, E\}$ of w, either this side is free in both H and H', or it is occupied either by an edge internal to G_ν or by an edge external to G_ν in both H and H'. This implies that, with the chosen orientation, for each edge e'' of $G \setminus G_\nu$ the relative position of the end-vertices of e'' in H'' is the same as in H. It follows that, with the chosen orientation, no edge of H'' is downward (Fig. 16). $\square$

Based on Theorem 4, in order to test RU planarity of a biconnected digraph G with a given reference edge on the external face, we exploit a dynamic programming technique that visits a rooted SPQR-tree T of G bottom-up. At each visited node ν of T, and for each RU-spirality τ_ν admitted by ν, we store at ν a pair $\langle \tau_\nu, H_\nu\rangle$, where H_ν is just one RU representation of G_ν with spirality τ_ν, called a *representative of* τ_ν. The set of all pairs $\langle \tau_\nu, H_\nu\rangle$ is the *feasible set of* ν and is denoted by Σ_ν. Observe that, if G_ν has n_ν vertices and if $\tau_\nu \in \Sigma_\nu$, the rectilinear spirality σ_ν in τ_ν cannot exceed n_ν, as we can make at most n_ν right or n_ν left turns. Also, for each value σ_ν, the number of RU spiralities τ_ν in Σ_ν with rectilinear spirality σ_ν is bounded by a constant. Hence, we have the following.

Property 2. *For any component G_ν with n_ν vertices, $|\Sigma_\nu| = O(n_\nu)$. Also, for each $\tau_\nu \in \Sigma_\nu$, the corresponding rectilinear spirality σ_ν belongs to the interval $[-n_\nu, n_\nu]$.*

Table 2
Relationships between the rectilinear spirality of an S- or P-component and the spirality values of their children [8]. In the third row of the table (P-node with two children), u and v denote the two poles of ν; also, for each pole $w \in \{u, v\}$, α_w^l (resp. α_w^r) is a binary variable such that $\alpha_w^l = 1$ (resp. $\alpha_w^r = 1$) if the leftmost (resp. rightmost) external angle at w in the P-component equals 90°, while $\alpha_w^l = 0$ (resp. $\alpha_w^r = 0$) if the leftmost (resp. rightmost) external angle at w in the P-component equals 180°; also, for each $d \in \{l, r\}$, $k_w^d = 1$ if the internal degree of w in μ_d and the external degree of w in ν equal 1; otherwise $k_w^d = \frac{1}{2}$ (refer to [8] for further details).

Type of node	Rectilinear Spirality Relationship
S-node ν with children μ_1, μ_2	$\sigma_\nu = \sigma_{\mu_1} + \sigma_{\mu_2}$
P-node ν with three children μ_l, μ_c, μ_r	$\sigma_\nu = \sigma_{\mu_l} - 2 = \sigma_{\mu_c} = \sigma_{\mu_r} + 2$
P-node ν with two children μ_l, μ_r	$\sigma_\nu = \sigma_{\mu_l} - k_u^l\alpha_u^l - k_v^l\alpha_v^l = \sigma_{\mu_r} + k_u^r\alpha_u^r + k_v^r\alpha_v^r$

6.2. Testing series-parallel digraphs in quadratic time

When the SPQR-tree T of a biconnected graph G does not have R-nodes, G is a *series-parallel graph*, or simply an *SP-graph*. Also, T is called the SPQ-tree of G. In this section we assume that G is an *SP-digraph*, i.e., a digraph whose underlying undirected graph is an SP-graph. We also assume that T is normalized. We prove the following lemmas.

Lemma 10. *Let ν be a Q-node of T. We can compute Σ_ν in $O(1)$ time.*

Proof. G_ν is a directed edge $e = (u, v)$ of G. In any RU representation of G, edge e is either leftward, or rightward, or upward. For each of these three possibilities, we have to consider the $O(1)$ possible arrangements of the edges adjacent to e on the different sides of u and v, each of them defining a different RU spirality τ_ν (each RU spirality has rectilinear spirality 0). Thus Σ_ν is constructed in $O(1)$ time. □

The next two lemmas show how to construct the feasible set of an S-node or of a P-node ν, provided that the feasible sets of the children of ν are known. In both cases, the idea is to check whether an RU representation of G_ν can be constructed by merging some representations of the pertinent digraphs of the children of ν. This merging is possible if there is compatibility between the pole side specifications of the children (the labels on each side must not be in contrast) and if the rectilinear spirality values of the children verify certain relationships, which are described in [8] and summarized in Table 2. In particular, in [8] it is proved that if ν is an S-node, the rectilinear spirality of a rectilinear representation of G_ν equals the sum of the rectilinear spirality values for the merged representations of the children of ν. If ν is a P-node, the spirality values of the children of ν differ for some additive constants, which depend on the chosen planar embedding for the parallel composition and on the number of children of ν (two or three children).

Lemma 11. *Let ν be an S-node of T with children μ_1 and μ_2, and let n_1^ν and n_2^ν be the number of nodes in G_{μ_1} and G_{μ_2}, respectively. If Σ_{μ_1} and Σ_{μ_2} are given, then we can compute Σ_ν in $O(n_1^\nu \cdot n_2^\nu)$ time.*

Proof. For each pair $\tau_{\mu_1} \in \Sigma_{\mu_1}$ and $\tau_{\mu_2} \in \Sigma_{\mu_2}$, let H_{μ_1} and H_{μ_2} be the representatives of τ_{μ_1} and τ_{μ_2}, respectively. Let u_i and v_i be the first-pole and second-pole of μ_i, respectively, with $i = 1, 2$. Clearly $v_1 = u_2$. Suppose that for each side $D \in \{S, N, W, E\}$ one of these three cases holds: (i) $D_{v_1} = \mathsf{free}$ in τ_{μ_1} and $D_{u_2} = \mathsf{free}$ in τ_{μ_2}; (ii) $D_{v_1} = \mathsf{int}$ in τ_{μ_1} and $D_{u_2} = \mathsf{ext}$ in τ_{μ_2}; (iii) $D_{v_1} = \mathsf{ext}$ in τ_{μ_1} and $D_{u_2} = \mathsf{int}$ in τ_{μ_2}. If so, the pole side specifications of τ_{μ_1} and τ_{μ_2} are compatible, and we can construct an RU representation H_ν by gluing together H_{μ_1} and H_{μ_2} at the common pole $v_2 = u_1$; also, we can easily compute the corresponding value of rectilinear spirality σ_ν in τ_ν based on σ_{μ_1}, σ_{μ_2}, and on the pole side specifications in τ_{μ_1} and τ_{μ_2}. In particular, the first-pole of ν is $u = u_1$, the second-pole of ν is $v = v_2$, and the pole side specifications φ_u and φ_v in τ_ν are inherited by those of τ_{μ_1} and τ_{μ_2}; the rectilinear spirality σ_ν equals $\sigma_{\mu_1} + \sigma_{\mu_2}$, as proved in [8]. Hence, the pair $\langle \tau_\nu, H_\nu \rangle$ is added to Σ_ν. Conversely, if for a side $D \in \{S, N, W, E\}$ none of the above conditions (i), (ii), and (iii) holds, then we discard the pair $\tau_{\mu_1}, \tau_{\mu_2}$, because their pole side specifications are not compatible, and hence H_{μ_1} and H_{μ_2} cannot be glued together.

About the time complexity, we have to consider all pairs of elements τ_{μ_1} and τ_{μ_2}, one taken in Σ_{μ_1} and the other taken in Σ_{μ_2}. By Property 2 these pairs are $O(n_1^\nu \cdot n_2^\nu)$, and for each pair we can check the compatibility of the two elements in $O(1)$ time. Also, when two elements are compatible and the corresponding representations H_{μ_1} and H_{μ_2} can be glued together, the resulting representative H_ν is encoded in constant time by suitably linking the descriptions of H_{μ_1} and H_{μ_2}. In conclusion, the whole procedure takes $O(n_1^\nu \cdot n_2^\nu)$ time. □

The next lemma is about the feasible sets of P-nodes. Theorem 5 summarizes the main result of this subsection.

Lemma 12. *Let ν be a P-node of T with children $\mu_1, \mu_2, \ldots \mu_h$ $(h = 2, 3)$. If Σ_{μ_1} and Σ_{μ_2} are given, then we can compute Σ_ν in $O(n)$ time.*

Proof. Let u and v be the first-pole and the second-pole of ν, respectively. By definition of P-node, u and v are also the first-pole and the second-pole of each child of ν. Denote by n_ν the number of vertices of G_ν. Suppose first that ν is a P-node with three children μ_1, μ_2, and μ_3. Any planar embedding of G defines a planar embedding of skel(ν). If for simplicity we topologically imagine v above u, in any given embedding of skel(ν) the three children of ν (namely, the edges of skel(ν) that correspond to these children) occur from left to right in some order (equivalently, this order coincides with the circular order in which these children are encountered around v moving counterclockwise from the reference edge of skel(ν)). Suppose that for a given embedding ϕ of skel(ν), we rename the three children of ν as μ_l, μ_c, and μ_r, if they occur in this left-to-right order in ϕ. Let H be any rectilinear representation of G that induces for skel(ν) the embedding ϕ. Also denote by σ_ν, σ_{μ_l}, σ_{μ_c}, and σ_{μ_r} the rectilinear spirality values of the restrictions of H to G_ν, G_{μ_l}, G_{μ_c}, and G_{μ_r}, respectively. It is proved in [8] that the following relationship holds: $\sigma_\nu = \sigma_{\mu_l} - 2 = \sigma_{\mu_c} = \sigma_{\mu_r} + 2$. Clearly, since an RU representation is in particular a rectilinear representation, then the same relationship must be verified for any RU representation that induces the embedding ϕ for skel(ν). Hence, as done in [8], for each candidate rectilinear spirality value $\sigma_\nu \in [-n_\nu, n_\nu]$ (see Property 2) and for each possible embedding ϕ of skel(ν), one can check in $O(1)$ time whether there exist three elements $\tau_{\mu_l} \in \Sigma_{\mu_l}$, $\tau_{\mu_c} \in \Sigma_{\mu_c}$, and $\tau_{\mu_r} \in \Sigma_{\mu_r}$ such that the corresponding rectilinear spiralities σ_{μ_l}, σ_{μ_c}, and σ_{μ_r} satisfy the above relationship. If not, then the target rectilinear spirality value σ_ν is not feasible, otherwise suppose that such elements τ_{μ_l}, τ_{μ_c}, and τ_{μ_r} exist. To check whether we can combine them into an RU-spirality τ_ν having rectilinear spirality σ_ν, we have to test the compatibility of the pole side specifications for each pole $w \in \{u, v\}$. This compatibility can be checked with the following simple considerations. Since ν has three children, w has degree four in G. Also, each of the three components G_{μ_l}, G_{μ_c}, and G_{μ_r} contains exactly one edge incident to w, while the fourth edge incident to w is external to all the three components. Hence, to have compatibility, we must have that in the pole specification of each τ_{μ_j} $(j = l, c, r)$ there is exactly one side of w with value int and each other side of w with value ext. In particular, call D_w the side of w in the pole specification of τ_{μ_l} for which $D_w =$ int. To fix the ideas, assume that $D_w \equiv W_w$, i.e., the edge of G_{μ_l} incident w occupies the West side of w (the cases $D_w \equiv S_w$, $D_w \equiv N_w$, and $D_w \equiv E_w$ are treated in a similar way). This means that $\varphi_w = (\text{ext}, \text{ext}, \text{int}, \text{ext})$ in τ_{μ_l}. Thus, in order to have compatibility at w it must be $\varphi_w = (\text{int}, \text{ext}, \text{ext}, \text{ext})$ in τ_{μ_c} and $\varphi_w = (\text{ext}, \text{ext}, \text{ext}, \text{int})$ in τ_{μ_r}. If there is compatibility at the pole u and at the pole v, we can glue together the representatives H_{μ_l}, H_{μ_c}, H_{μ_r} into an RU representation H_ν with rectilinear spirality σ_ν, and we insert $\tau_\nu = \langle \sigma_\nu, H_\nu \rangle$ in Σ_ν; otherwise we discard the triplet $\tau_{\mu_l}, \tau_{\mu_c}, \tau_{\mu_r}$.

The described procedure for constructing Σ_ν takes $O(n)$ time because: (i) by Property 2 there are $O(n)$ possible target rectilinear spirality values to consider; (ii) for each target spirality value, skel(ν) has 6 distinct planar embeddings to consider; (iii) for each embedding of skel(ν) we can check in $O(1)$ time which triplets $\tau_{\mu_l}, \tau_{\mu_c}, \tau_{\mu_r}$ that satisfy the relation $\sigma_\nu = \sigma_{\mu_l} - 2 = \sigma_{\mu_c} = \sigma_{\mu_r} + 2$ (see also [8]), and for each of these triplets we can also check in $O(1)$ time the compatibility of the pole side specifications.

If ν is a P-node with two children, the strategy for constructing Σ_ν is exactly the same. However in this case, skel(ν) has only two embeddings to consider for each target value of rectilinear spirality σ_ν. Also, for each of these two embeddings, the relationship between σ_ν and the rectilinear spiralities of the children of ν, as well as the compatibility conditions for the pole side specifications, may require to analyze more cases whose number is however still bounded by a constant (see [8] for details about the relationships between a rectilinear representation of ν and those of its two children). $\square$

The next lemma shows how to perform the analysis described in Lemmas 10 to 12 so that the time complexity of computing the feasible set of all the nodes overall the rooted SPQR-trees is $O(n^2)$. The strategy that we consider is similar to the one described in [20,27].

Lemma 13. *There exists an algorithm that computes the feasible sets of all nodes over all the rooted SPQ-trees of G in $O(n^2)$ time.*

Proof. Let $E = \{e_1, \dots, e_m\}$ be the set of edges of G. For each edge $e_i \in E$, denote by T_i the SPQ-tree of G rooted at the Q-node corresponding to e_i. By Lemma 10, we can compute the feasible set of each Q-node in $O(1)$ time; since there are $O(n)$ Q-nodes in each tree T_i and since $m = O(n)$, the feasible sets of all Q-nodes over all sequence $T_1, \dots, T_m$ is computed in $O(n^2)$ time. We now show the statement for the P-nodes and for the S-nodes. We will denote by Σ_ν^i the feasible set of a node ν in T_i, where $i \in \{1, \dots, m\}$.

P-nodes Let T_i be the currently visited tree and let ν be a P-node of T_i. If the parent of ν in T_i coincides with the parent of T_j for some $j \in \{1, \dots, i-1\}$, and if Σ_ν^j was previously computed, then the algorithm does not need to recompute Σ_ν^i, because $\Sigma_\nu^i = \Sigma_\nu^j$. Hence, for each P-node ν, the number of computations of its feasible set that are performed over all possible trees is at most 4 (one for each different choice of the parent of ν). For a P-node ν whose feasible set needs to be computed for the first time in T_i (it is always the case if $i = 1$), computing Σ_ν^i takes $O(n)$ time by Lemma 12. Hence, since the SPQ-tree contains $O(n)$ P-nodes in total and since the feasible set of each P-node is computed at most four times over all T_i $(i \in \{1, \dots, h\})$, the time needed to compute the feasible sets of all P-nodes over all sequence of rooted SPQ-trees is $O(n^2)$.

S-nodes To show the statement for the S-nodes, we exploit two claims, namely Claim 1 and Claim 2. Given an S-node ν of T_1, let n_1^ν and n_2^ν denote the number of vertices in the pertinent graphs of its two children. We prove the following.

Claim 1. *Let $\mathcal{S}$ be the set of all S-nodes in T_1. We have $\sum_{\nu\in\mathcal{S}} n_1^\nu \cdot n_2^\nu = O(n^2)$.*

Let ξ be any node of T_1. Denote by $T_1(\xi)$ the subtree of T_1 rooted at ξ, and let $\mathcal{S}(\xi)$ be the set of S-nodes in $T_1(\xi)$. Denote by $s(\xi) = \sum_{\nu\in\mathcal{S}(\xi)} n_1^\nu \cdot n_2^\nu$. Also, let n_ξ and m_ξ be the number of vertices and the number of edges of G_ξ, respectively. We prove that $s(\xi) \le 4m_\xi^2$ by induction on the depth d of $T_1(\xi)$. In the base case $d = 0$ and ξ is a Q-node (i.e., it is a leaf); we have $s(\xi) = 0 < m_\xi$. In the inductive case, $d \ge 1$ and we assume (by the inductive hypothesis) that the property holds for every node in the subtree of $T_1(\xi)$. There are two cases, i.e., either ξ is an S-node or ξ is a P-node. Suppose first that ξ is an S-node, and let μ_1 and μ_2 be its children. We have $s(\xi) = n_{\mu_1}n_{\mu_2} + s(\mu_1) + s(\mu_2)$. By the inductive hypothesis and since $n_{\mu_i} \le m_{\mu_i} + 1$ $(i \in \{1, 2\})$, we get $s(\xi) \le m_{\mu_1}m_{\mu_2} + m_{\mu_1} + m_{\mu_2} + 1 + 4m_{\mu_1}^2 + 4m_{\mu_2}^2 \le 4(m_{\mu_1} + m_{\mu_2})^2$. Since $m_{\mu_1} + m_{\mu_2} = m_\xi$, we have $s(\xi) \le 4m_\xi^2$. Suppose now that ξ is a P-node, and let $\mu_1, \dots, \mu_k$ be the children of ξ, with $k \in \{2, 3\}$. We have $s(\xi) = s(\mu_1) + \dots + s(\mu_k)$. By inductive hypothesis and since $m_{\mu_1} + \dots + m_{\mu_k} = m_\xi$, we get $s(\xi) \le 4m_{\mu_1}^2 + \dots + 4m_{\mu_k}^2 \le 4(m_{\mu_1} + \dots + m_{\mu_k})^2 = 4m_\xi^2$. If ξ is the root child ot T_1 then $\mathcal{S}(\xi)$ is the set of all the S-nodes of T_1; hence Claim 1 holds because $m_\xi = O(n)$.

For an S-node ν of T_i, denote by n_0^ν the number of vertices incident to the edges of G that are not in the pertinent graph of ν. Also, let $\mu_1(\nu)$ and $\mu_2(\nu)$ be the two children of ν in T_i, where n_1^ν and n_2^ν denote the number of vertices of their pertinent graphs. We prove the following.

Claim 2. *The feasible set Σ_ν^i can be computed in $O(\min\{n_1^\nu \cdot n_2^\nu, n_1^\nu \cdot n_0^\nu, n_2^\nu \cdot n_0^\nu\})$ time if the feasible sets of the children of ν are given.*

Suppose first that $n_0^\nu = \max\{n_0^\nu, n_1^\nu, n_2^\nu\}$. In this case, by Lemma 11, we can compute Σ_ν^i in $O(n_1^\nu \cdot n_2^\nu) = O(\min\{n_1^\nu \cdot n_2^\nu, n_1^\nu \cdot n_0^\nu, n_2^\nu \cdot n_0^\nu\})$. Suppose vice versa that $\max\{n_0^\nu, n_1^\nu, n_2^\nu\}$ is one among n_1^ν and n_2^ν, say for example $n_2^\nu = \max\{n_0^\nu, n_1^\nu, n_2^\nu\}$ (if the maximum is n_1^ν, the argument is analogous). The rectilinear spirality values admitted by ν must be in the interval $[-(n_0^\nu + 4), +(n_0^\nu + 4)]$, because the number of right turns minus the number of left turns walking counterclockwise on the boundary of any cycle of a rectilinear representation of G equals 4, and because any rectilinear representation of G restricted to $G \setminus G_\nu$ cannot have more than n_0 turns in the same direction (either left or right). Also, recall that the rectilinear spirality values admitted by ν are either all integer or all semi-integer numbers, depending on the in-degree and out-degree of the poles of ν. Hence, to construct the feasible set Σ_ν^i, we can consider every pair $\{\tau_\nu, \tau_1\}$, with τ_ν being a candidate RU-spirality for ν such that the corresponding rectilinear spirality σ_ν is in the interval $[-(n_0^\nu + 4), +(n_0^\nu + 4)]$ and $\tau_1 \in \Sigma_{\mu_1}^i$. For each such pair we check whether there exists a value $\tau_2 \in \Sigma_{\mu_2}^i$ such that: (*i*) $\sigma_1 + \sigma_2 = \sigma_\nu$, where σ_1 and σ_2 are the rectilinear spiralities of τ_1 and τ_2, respectively; (*ii*) the pole side specifications of τ_1 and τ_2 are compatible (see the considerations in the proof of Lemma 11). In the positive case, τ_ν is inserted in Σ_ν^i, otherwise τ_ν is discarded. There are $O(n_0^\nu \cdot n_1^\nu)$ distinct pairs $\{\tau_\nu, \tau_1\}$. For each pair we can check in $O(1)$ time whether there exists a value τ_2 that satisfying the conditions above. In fact, there are $O(1)$ elements in $\Sigma_{\mu_2}^i$ with rectilinear spirality σ_2. Also, we can construct every feasible sets as a dictionary data structure that, for each key σ, returns all the RU-spirality with rectilinear spirality σ. Hence, this procedure takes $O(n_0^\nu \cdot n_1^\nu) = O(\min\{n_1^\nu \cdot n_2^\nu, n_1^\nu \cdot n_0^\nu, n_2^\nu \cdot n_0^\nu\})$ time.

We are now ready to prove the statement of the lemma for the S-nodes. Let T_i be the currently visited tree and let ν be an S-node. As for the P-nodes, if the parent of ν in T_i coincides with the parent of T_j for some $j \in \{1, \dots, i-1\}$, and if Σ_ν^j was previously computed, then the algorithm does not need to compute Σ_ν^i, because $\Sigma_\nu^i = \Sigma_\nu^j$. Hence, for each S-node ν, the number of computations of its feasible sets that are performed over all possible trees is at most 3 (one for each different choice of the parent of ν). Suppose that, for an S-node ν, μ_1 and μ_2 are the children of ν in the first rooted tree T_1, and that n_1^ν and n_2^ν are the number of vertices of G_{μ_1} and G_{μ_2}, respectively. Computing the feasible set in T_1 of ν costs $O(n_1^\nu \cdot n_2^\nu)$ by Lemma 11. By Claim 2, computing the feasible set of an S-node ν in each tree T_i takes a time that is asymptotically not greater than the one spent in T_1, i.e., $O(n_1^\nu \cdot n_2^\nu)$. Denote by $\mathcal{S}$ the set of all S-nodes in T_1. Since the feasible set of each S-node has to be computed at most three times over all T_i $(i = 1, \dots, h)$, the time required to compute the feasible sets of all S-nodes over all trees is $O(\sum_{\nu\in\mathcal{S}} n_1^\nu \cdot n_2^\nu)$, which, by Claim 1, is $O(n^2)$. □

Theorem 5. *Let G be an n-vertex SP-digraph. There exists an $O(n^2)$-time algorithm that tests whether G admits an RU representation, and that computes one in the affirmative case.*

Proof. Let T be the SPQ-tree of G. For each Q-node ρ of T, the algorithm considers T rooted at ρ and tests whether G admits an RU representation with the reference edge G_ρ on the external face (recall that this rooted tree describes all planar embeddings of G with G_ρ on the external face). The testing algorithm performs a post-order visit of T. It first computes Σ_ν for each leaf ν of T, that is, for each Q-node of T distinct from ρ. This can be done in $O(n)$ time by Lemma 10. Then, for each internal node ν of T distinct from ρ the algorithm computes Σ_ν by using the feasible sets of the children of ν, by means of Lemma 11 or of Lemma 12 depending on whether ν is an S-node or a P-node. If Σ_ν is empty, then G does not have an RU representation with G_ρ on the external face. In this case the algorithm stops visiting T rooted at ρ, and starts visiting T rooted at another Q-node. Suppose vice versa that the algorithm achieves the root child ν and that Σ_ν is not empty. In this case, it is sufficient to check whether there exists an element $\tau_\nu \in \Sigma_\nu$ whose H_ν can be glued together with a straight-line representation of the reference edge G_ρ, which is oriented either upward, or leftward, or rightward. This can be easily done by checking: (i) the compatibility with the pole side specifications of τ_ν; for example, if G_ρ is oriented upward (the cases leftward and rightward are similar), then in τ_ν it must be $N_u = \text{ext}$ and $S_v = \text{ext}$; and (ii) by checking that the rectilinear spirality σ_ν in τ_ν guarantees that the number of right turns minus the number of left turns encountered traversing clockwise any arbitrarily chosen cycle of G passing through the reference edge G_ρ equals four (see also [1,45] for details about how to check this condition on the rectilinear spirality).

The $O(n^2)$-time complexity of the testing algorithm immediately follows from Lemma 13 and from the $O(n)$-time procedure described above for checking the condition at the root of each of the $O(n)$ rooted SPQ-trees. $\square$

6.3. FPL testing algorithm by the number of sources and sinks

We assume that G is a biconnected planar digraph and T is a (normalized) rooted SPQR-tree of G. A vertex of G that is either a source or a sink of G is called a *switch* [1]. In this section we give an FPT algorithm for Rectilinear-Upward Planarity Testing parameterized by the number k of switches of G.

Overview of the algorithm Let ν be any node of T. As we will show in Lemma 14, the following hold:

- If G_ν does not contain any switches of G, then it can only admit $O(1)$ rectilinear spirality values, and hence a number of RU-spirality values bounded by a constant;
- Else, the possible values of rectilinear spirality admitted by G_ν is a linear function of k.

The FPT algorithm extends the dynamic programming technique of Section 6.2 so to handle R-nodes in addition to S-, P-, and Q-nodes. When an R-node ν of a rooted SPQR-tree is visited, we consider the two possible planar embeddings of its skeleton, and for each of these two embeddings we consider every possible upward planar embedding and every possible target RU-spirality τ_ν; then, we test whether G_ν admits τ_ν. If so, as for S-nodes and P-nodes, we construct an RU representation H_ν and we insert $\langle \tau_\nu, H_\nu \rangle$ in Σ_ν; otherwise, τ_ν is discarded. H_ν is encoded in a time and in a space that are linear in the size of skel(ν), by suitably linking the representatives for the children of ν. To perform the test for each target value τ_ν, we partition the children of ν into two sets A and B. A child μ of ν is inserted in A if G_μ contains at least one switch of G, otherwise we insert μ in B. Clearly $|A| = O(k)$, whereas the size of the feasible set of any element in B is bounded by a constant. Then, for each combination of fixed RU-spirality values in the feasible sets of the elements in A, we solve a constrained RU planarity testing problem that: (a) forces G_ν to have the target rectilinear spirality σ_ν associated with τ_ν; (b) preserves the chosen combination of RU-spirality values for the elements of A; and (c) guarantees that the pertinent digraphs of the nodes in B have one of the constantly many RU-spirality values in their feasible sets. Denoted by $n_{\text{skel}(\nu)}$ the number of vertices in skel(ν), we prove that this test can be executed in $O(n_{\text{skel}(\nu)})$ time by using the 2-SAT model of Section 5 enriched with an $O(n_{\text{skel}(\nu)})$ number of constraints. The problem belongs to the FPT class when parameterized by k as a consequence of the following facts: (i) there are $O(k^k) = O(2^{k \log k})$ combinations of RU-spirality values for the elements in A; (ii) there are $O(3^k) = O(2^{2k})$ upward planar embeddings for each of the two possible planar embeddings of an R-node (each switch has degree at most three, otherwise the digraph cannot have an RU representation, and there are at most three different left-to-right orderings of its incident edges for any given fixed circular ordering); and (iii) $\sum_{\nu \in T} n_{\text{skel}(\nu)} = O(n)$.

Bounds on the rectilinear spirality In the remainder of this section, a *reflex* angle is a 270° angle and an *inflex* angle is a 90° angle. The following lemma will be used to prove Theorem 6. The lemma establishes that the absolute value of rectilinear spirality of an RU representation H_ν, for any node ν of T, is bounded by a function that is linear in the number of switches of G that belong to G_ν. The intuition behind this property is that the increase of the absolute value of rectilinear spirality of a component requires an increase of switches in the external face to avoid downward edges.

Lemma 14. *Let H be an RU representation of G, let ν be a node of T, and let σ_ν be the rectilinear spirality of the restriction H_ν of H to G_ν. If k_ν denotes the number of switches of G belonging to G_ν, then $\sigma_\nu \in [-2k_\nu - 3, 2k_\nu + 3]$.*

Proof. We prove the lemma for $\sigma_\nu \geq 0$ (the case $\sigma_\nu \leq 0$ can be proved with a symmetric argument). Refer to Fig. 17. Let u and v be the first-pole and the second-pole of ν, respectively. Let P_l be the left path of the external face of H_ν connecting u to v. Let $a = n(P_l^{uv})$ be the number of right turns minus left turns of P_l while moving from u to v. By definition of rectilinear spirality, we have $\sigma_\nu \leq a$. We can assume that $a \geq 3$, otherwise the statement trivially holds. Let H_l be the restriction of H_ν to P_l. An example is given in Fig. 17a; it depicts a component H_ν where H_l is highlighted, and we have $\sigma_\nu = \frac{19}{2}$ and $a = 11$. A subpath $\tilde{P}_l$ of P_l is *0-horizontal* (resp. *0-vertical*) if the following conditions hold: (i) each of its two endpoints w_1 and w_2 is either a degree-2 vertex in H_ν forming a reflex angle in the external face of H_ν or it is a pole of ν; (ii) $n(\tilde{P}^{w_1 w_2}) = 0$; (iii) the edges of $\tilde{P}_l$ incident to w_1 and to w_2 are both horizontal (resp. vertical) in H_ν. A 0-horizontal (resp. 0-vertical) subpath $\tilde{P}_l$ is *maximal* if it is not a subpath of any other 0-horizontal (resp. or 0-vertical) subpath. In Fig. 17a the subpath of P_l connecting w_1 and w_2 is 0-horizontal, but it is not maximal since it is contained in the 0-horizontal subpath of P_l connecting v_{11} to v. Let $\{u = v_0, v_1, \ldots, v_b, v = v_{b+1}\}$ be the subset of vertices of H_ν such that:

- $v_1, \ldots, v_b$ are degree-2 vertices of H_ν;

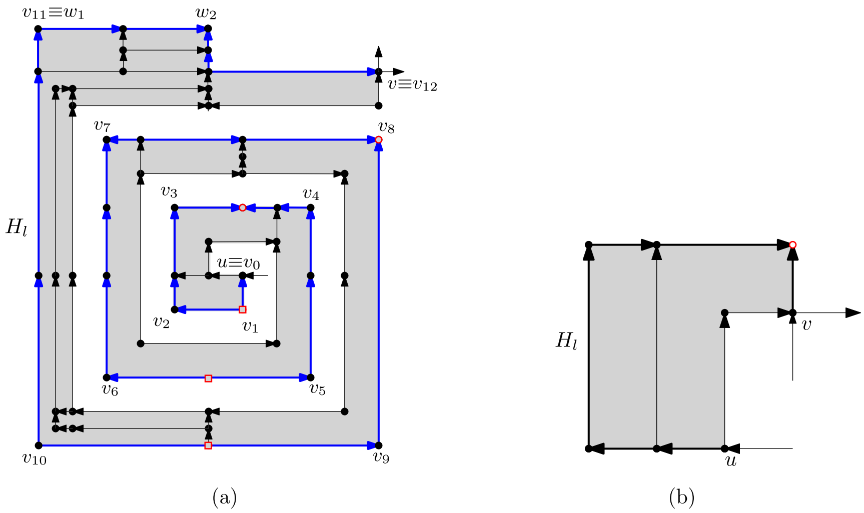


Fig. 17. Illustration for the proof of Lemma 14.

- while traversing P_l from u to v, we encounter the vertices $v_1, \ldots, v_b$ in this order;
- for each $i \in [0, b]$, the path $P_i \subseteq P_l$ having v_i and v_{i+1} as endpoints is either a maximal 0-horizontal or a maximal 0-vertical path.

The latter property implies $P_0 \cup \ldots \cup P_b = P_l$. Also, since for each path P_i ($i \in [0, b]$) we have $n(P_i^{v_i v_{i+1}}) = 0$, it follows that $b = a$. Since paths $P_1, \ldots, P_{a-1}$ are maximal, we have that 0-horizontal and 0-vertical paths alternate, and hence at least $\lfloor \frac{a-1}{2} \rfloor \geq \frac{a-1}{2} - 1$ of these paths are 0-horizontal. Let P_i be any 0-horizontal path such that $1 \leq i \leq a-1$. We have that all the vertices of $G_\nu \setminus P_i$ that are adjacent to some vertex of P_i are either below P_i (Case 1) or above P_i (Case 2). Also, since v_i and v_{i+1} are incident to vertical edges, we have that P_i contains at least one sink of G in Case 1, while it contains at least one source of G in Case 2. In Fig. 17a the sources contained in the 0-horizontal paths P_1, P_5, and P_9 are represented as empty (red) squares, while the sinks in paths P_3 and P_7 are represented as empty (red) circles. Hence, $k_\nu \geq \frac{a-1}{2} - 1$, $a \leq 2k_\nu + 3$, and $\sigma_\nu \in [0, 2k_\nu + 3]$. $\square$

Note that the spirality σ_ν can be larger than 2, even if G_ν contains only one switch of G. For example, Fig. 17b shows an RU representation of H_ν that contains only one switch of G (red empty circle) and $\sigma_\nu = \frac{5}{2}$.

From now on we assume that G has at least one source and one sink, otherwise G cannot have an upward drawing and we can immediately reject the instance. Also, recall that k denotes the number of switches of G. When we visit an R-node ν of T and we consider a target RU-spirality τ_ν, for each fixed combination of RU-spirality values of the children in A and for each one of the two planar embeddings of skel(ν), we have to test whether τ_ν can be inserted in the feasible set of ν. To efficiently perform this test, for each child μ of ν that corresponds to a virtual edge of skel(ν) we substitute its virtual edge in skel(ν) with a suitable *gadget*, consisting of a constant-size digraph. If μ belongs to A then this gadget will have a fixed RU representation, which reflects the selected RU-spirality of its corresponding component in the feasible set of μ. Namely, the gadget for μ is a succinct representation of the representative of G_μ for the selected RU-spirality, which we call *compressed RU representation*. If μ belongs to B then the gadget will have some forbidden RU representations depending on the feasible set of μ. Again, this gadget is a succinct representation of G_μ, but it does not have a fixed RU representation. The digraph obtained from skel(ν) after the replacement of each virtual edge with a corresponding gadget is denoted as $\tilde{G}_\nu$. Once we have tested the existence of an RU-representation $\tilde{H_\nu}$ for $\tilde{G}_\nu$ (with the necessary constraints), we can use Theorem 4 to derive from $\tilde{H_\nu}$ a corresponding RU-representation H_ν of G_ν, by substituting each gadget with an RU representation of its original subgraph having the same RU-spirality. In the following we describe the types of gadgets for the children of ν.

Compressed RU representations for the elements in A. Let μ be a child of ν such that $\mu \in A$. Let u and v be the poles of μ and suppose that we are given an RU-spirality $\tau = \langle \sigma, \varphi_u, \varphi_v \rangle$ for μ. Assume that $G(\tau)$ is a plane digraph such that: (*i*) it contains the vertices u and v, and these two vertices are on the external face of $G(\tau)$; (*ii*) the (directed) edges of $G(\tau)$ incident to u (resp. to v) are in one-to-one correspondence with the (directed) edges of G_μ incident to u (resp. to v). Also, let $H(\tau)$ be any RU representation of $G(\tau)$ with RU-spirality τ. We say that $H(\tau)$ is a τ-*representative* for μ. Further, We say that $H(\tau)$ is *compressed* if $G(\tau)$ has $O(\sigma)$ vertices. Observe that any representation H_μ of G_μ having RU-spirality τ is a τ-representative for μ, but it may not be compressed.

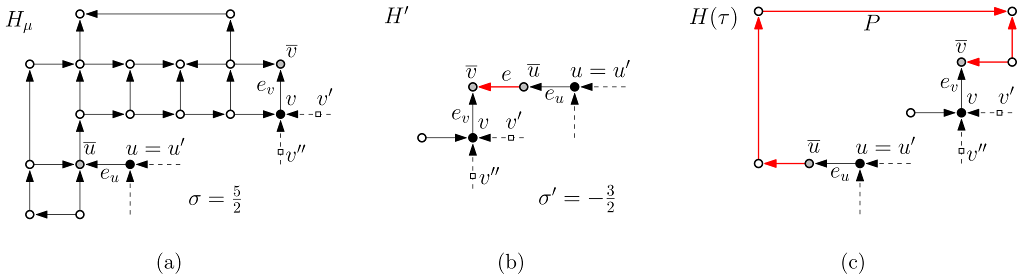


Fig. 18. Illustration of the procedure described in Lemma 15 to construct a compressed τ-representative for a component G_μ: (a) A representative H_μ of G_μ with RU-spirality $\tau = \langle \sigma = \frac{5}{2}, \varphi_u = (\text{int, free, int, ext}), \varphi_v = (\text{ext, int, int, ext}) \rangle$. Since u has degree 1 in H_μ, the alias vertex u' of u coincides with u; since v has degree 2 in H_μ and external degree 2, v has two alias vertices (small squares) v' and v'', on the two external edges (dashed). (b) An initial representation H' with RU-spirality $\tau' = \langle \sigma' = -\frac{3}{2}, \varphi_u = (\text{int, free, int, ext}), \varphi_v = (\text{ext, int, int, ext}) \rangle$, obtained by keeping the poles and their neighbors, and by connecting with a horizontal edge e two arbitrarily chosen neighbors $\overline{u}$ and $\overline{v}$ of u and v, respectively. (c) A final compressed τ-representative $H(\tau)$ for μ obtained by replacing e with a path P having $\sigma - \sigma' = 4$ degree-2 vertices that correspond to right turns while going from $\overline{u}$ to $\overline{v}$.

Lemma 15. *Let ν be an R-node of T, let μ be a child of ν such that $\mu \in A$, and let u and v be the first-pole and second-pole of μ, respectively. Let $\tau = \langle \sigma, \varphi_u, \varphi_v \rangle$ be an RU-spirality for μ. Given any representation H_μ of G_μ with RU-spirality τ, it is possible to construct in $O(\sigma)$ time a compressed τ-representative for μ.*

Proof. To construct a compressed τ-representative $H(\tau)$ of μ from H_μ we use the following algorithm. Refer to Fig. 18 for an illustration. First, $H(\tau)$ has the same poles u and v as H_μ and exactly the set of edges incident to u and v in H_μ (together with their end-vertices); for each pole $w \in \{u, v\}$, every edge incident to w enters or leaves w from the same side as in H_μ and it is oriented as in H_μ. Also, since $H(\tau)$ must have RU-spirality τ, we make the configurations of the external edges at the poles u and v coinciding with those determined by τ. At this point, we arbitrarily choose any neighbor $\overline{u}$ of u and any neighbor $\overline{v}$ of v. If $\overline{u}$ and $\overline{v}$ coincide, the construction of $H(\tau)$ is complete, and it is clear that $H(\tau)$ has RU-spirality τ; furthermore, the construction of $H(\tau)$ required $O(1)$ time and $H(\tau)$ has size $O(1)$. Otherwise, we add to $H(\tau)$ a suitable rectilinear path P from $\overline{u}$ to $\overline{v}$, in such a way that the rectilinear spirality of $H(\tau)$ equals σ. This is done with the strategy described below.

Preliminarly, connect $\overline{u}$ and $\overline{v}$ with a straight (horizontal or vertical) edge e; then, obtain P by possibly subdividing e with a suitable number of degree-2 vertices, corresponding all to right turns or all to left turns, so that $H(\tau)$ gets rectilinear spirality σ. More precisely, let e_u be the edge connecting u to $\overline{u}$, and let e_v be the edge connecting v to $\overline{v}$. If e_u and e_v are both horizontal and if u and v are both to the left or both to the right of $\overline{u}$ and of $\overline{v}$, respectively, then e must be vertical. If e_u and e_v are both vertical and if u and v are both to the bottom or both to the top of $\overline{u}$ and of $\overline{v}$, respectively, then e must be horizontal. In all other cases, e can be arbitrarily chosen as a horizontal or as a vertical edge. In particular, if e is vertical it is oriented upward, otherwise e can be arbitrarily oriented leftward or rightward. After the addition of e, the representation H' obtained so far is connected and we can measure its rectilinear spirality σ'. If $\sigma' = \sigma$, then we let $H(\tau) = H'$ and it is clear that $H(\tau)$ has size $O(1)$ and has been computed in $O(1)$ time. Suppose vice versa that $\sigma' \neq \sigma$, and let $\delta = \sigma - \sigma'$. Since the configurations of the (internal and external) edges incident to u and v are the same in $H(\tau)$ and H_μ, $|\delta|$ is a multiple of 4. If $\delta > 0$ then we obtain $H(\tau)$ from H' by replacing e with a path P consisting of δ degree-2 vertices and create a right turn at each of these vertices, while going from $\overline{u}$ to $\overline{v}$. Since δ is a multiple of 4, the first (resp. the last) edge of P is incident to $\overline{u}$ (resp. to $\overline{v}$) from the same side as e. Each vertical edge of P can be oriented upward while each horizontal edge of P can be oriented either leftward or rightward. If $\delta < 0$ we follow a symmetric procedure, by subdividing e with $-\delta$ degree-2 vertices and by creating a left turn at each of these vertices. This guarantees that the rectilinear spirality of the obtained representation $H(\tau)$ equals σ, and therefore $H(\tau)$ has RU-spirality τ. Since $\delta = O(\sigma)$ then $H(\tau)$ has $O(\sigma)$ size and is constructed in $O(\sigma)$ time. □

Gadgets for the elements in B. Let μ be a child of ν such that $\mu \in B$. The gadget that in $\tilde{G}_\nu$ corresponds to G_μ is defined based on the number of alias vertices associated with the poles of G_μ in G. Let u and v be the poles of G_μ and assume without loss of generality that u is the source and v is the sink of G_μ. Observe that u and v cannot be both sources or both sinks of G_μ, as otherwise G_μ would contain at least one switch of G, by contradicting the hypothesis that μ belongs to the set B. We distinguish the following cases (see Fig. 19):

- G_μ has exactly one alias vertex associated with u and exactly one alias vertex associated with v. In this case, the two alias vertices coincide with u and v, because skel(μ) is triconnected and hence u and v have at least two incident edges outside G_ν, whereas they have degree one in G_μ. The gadget corresponding to G_μ in $\tilde{G}_\nu$ is a path of length three oriented from u to v (see Fig. 19a). We call it a Type 1 gadget.
- G_μ has exactly one alias vertex associated with u (coincident with u) and two alias vertices at v. The gadget corresponding to G_μ in $\tilde{G}_\nu$ is the graph depicted in Fig. 19b. It consists of a series composition of an edge (u, w) and of a parallel of two paths of length three oriented from w to v. We call this gadget a Type 2 gadget.

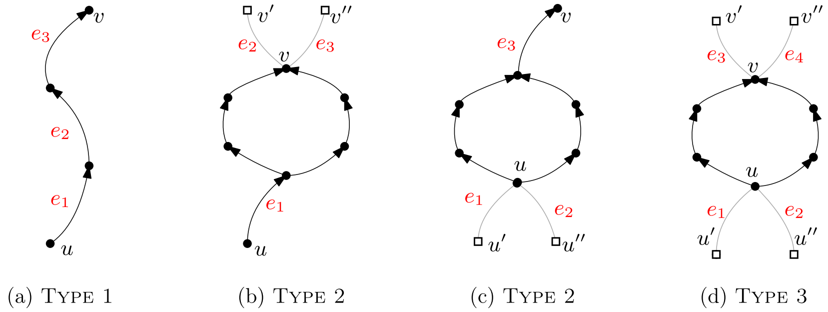


Fig. 19. The types of gadgets for the children of an R-node that belong to the set B.

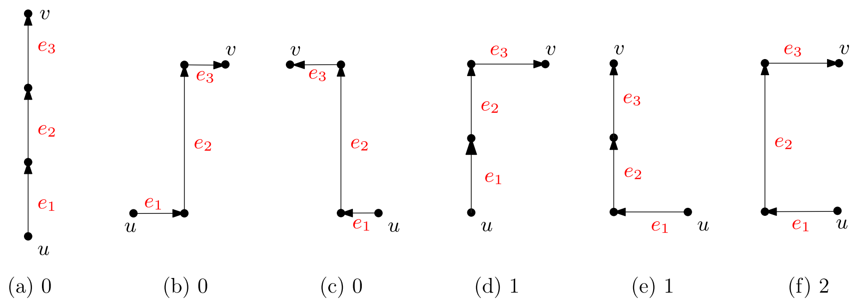


Fig. 20. A subset of RU representations for a Type 1 gadget that covers the possible RU-spirality values with non-negative rectilinear spirality (i.e., 0, 1, 2). Those with negative rectilinear spirality values (i.e., -1, -2) are symmetric to (d), (e), (f).

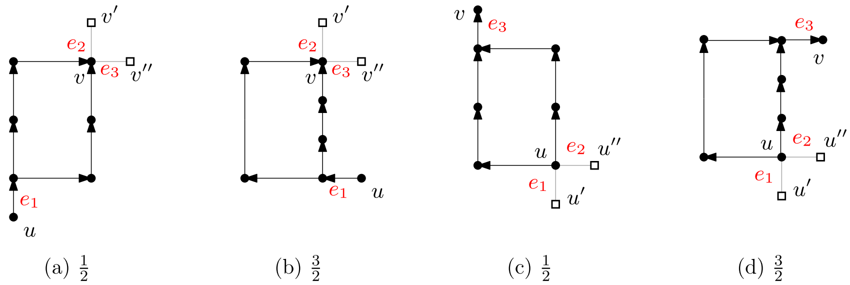


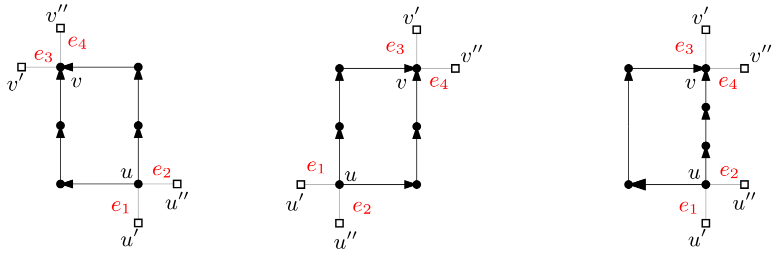
Fig. 21. A subset of RU representations for a Type 2 gadget that covers the possible RU-spirality values with non-negative rectilinear spirality (i.e., $\frac{1}{2}$, $\frac{3}{2}$). Those with negative rectilinear spirality values (i.e., $-\frac{1}{2}$, -$\frac{3}{2}$) are symmetric.

- G_μ has exactly one alias vertex associated with v (coincident with v) and two alias vertices at u. The definition of the gadget is symmetric to the previous case. Again, we call this gadget a Type 2 gadget; see Fig. 19c.
- G_μ has two alias vertices associated with u and two alias vertices associated with v: The gadget is a parallel of two paths of length three oriented from u to v. It is called a Type 3 gadget; see Fig. 19d.

Figs. 20 to 22 show subsets of RU representations of the gadgets defined above that cover their possible RU-spirality values. Given a node $\mu \in B$, denote by $T(\mu)$ the subtree rooted at μ in the rooted SPQR-tree T (i.e., $T(\mu)$ consists of μ and its descendants). Denote by $\tilde{\mu}$ a node such that if we replace $T(\mu)$ with $T(\tilde{\mu})$ in T, then the pertinent digraph $G_{\tilde{\mu}}$ is the gadget corresponding to G_μ in $\tilde{G_\nu}$. The following lemma proves that the RU-spirality set of $\tilde{\mu}$ is a superset of the RU-spirality set of μ.

Lemma 16. *Let μ be a child of ν such that $\mu \in B$. If $\tau \in \Sigma_\mu$ then $\tau \in \Sigma_{\tilde{\mu}}$.*

Proof. Let u and v be the first and the second pole of μ, and let H_μ be an RU representation of G_μ with RU-spirality τ. Assume that u is the source of G_μ and that v is the sink of G_μ (the argument is symmetric in the opposite case). Let P_l and P_r be the left and the right paths of the external face of H_μ while moving from u to v. Note that these two paths may share some vertices and edges. For each path $P \in \{P_l, P_r\}$, let σ_P be the difference between right and left turns along P in H_μ while moving from u to v. Denote by $D_u^P \in \{S, N, E, W\}$ the side of u to which P is incident. Similarly, denote by $D_v^P \in \{S, N, E, W\}$ be the side of v to which P is incident.



(a) 0 (b) 0 (c) 1

Fig. 22. A subset of RU representations for a TYPE 3 gadget that covers the possible RU-spirality values with non-negative rectilinear spirality (0, 1). An RU representation with rectilinear spirality -1 is symmetric to (c).

Observe that in any RU drawing of H_μ there is no vertex whose y-coordinate is smaller than the y-coordinate of u, otherwise one of such vertices would be a source of G. Similarly, in any RU drawing of H_μ there is no vertex whose y-coordinate is larger than the y-coordinate of v, otherwise one of such vertices would be a sink of G. By the considerations above, we have that the following two properties hold: (a) For each $P \in \{P_l, P_r\}$, $|\sigma_P| \leq 2$. (b) The edges of P_l and P_r incident to u are not incident to the South side of u (i.e., $D_u^P \neq S$) and the edges of P_l and P_r incident to v are not incident to the North side of v (i.e., $D_v^P \neq N$). The above two properties of H_μ imply the following possible cases for σ_P, D_u^P, and D_v^P in H_μ (with $P \in \{P_l, P_r\}$):

- $\sigma_P = -2$, $D_u^P = E$, $D_v^P = E$;
- $\sigma_P = -1$, $D_u^P = N$, $D_v^P = E$;
- $\sigma_P = -1$, $D_u^P = E$, $D_v^P = S$;
- $\sigma_P = 0$, $D_u^P = E$, $D_v^P = W$;
- $\sigma_P = 0$, $D_u^P = N$, $D_v^P = S$;
- $\sigma_P = 0$, $D_u^P = W$, $D_v^P = E$;
- $\sigma_P = 1$, $D_u^P = N$, $D_v^P = W$;
- $\sigma_P = 1$, $D_u^P = W$, $D_v^P = S$;
- $\sigma_P = 2$, $D_u^P = W$, $D_v^P = W$;

For each of the cases above, each path $\tilde{P}$ between u and v in the gadget $G_{\tilde{\mu}}$ can be represented as P, i.e., with the same difference between right and left turns as P and so that $\tilde{P}$ is incident to the sides of u and v as P. Also, thanks to how the different types of gadgets are defined, for each pole $w \in \{u, v\}$ the (directed) edges incident to w in $G_{\tilde{\mu}}$ are in one-to-one correspondence with the (directed) edges incident to w in G_μ. Hence, it is always possible to construct an RU representation $H_{\tilde{\mu}}$ of $G_{\tilde{\mu}}$ having the same rectilinear spirality and the same pole side specifications as in τ, i.e., an RU representation $H_{\tilde{\mu}}$ with RU-spirality τ. This proves that $\tau \in \Sigma_{\tilde{\mu}}$. □

As we said, Lemma 16 shows that the gadget $G_{\tilde{\mu}}$ for an element μ of B can take all RU spirality values admitted by μ in an RU representation of G. However, the reverse in not true in general, that is, there might be some RU-spirality values admitted by $\tilde{\mu}$ that are not admitted by μ. For instance, if $G_{\tilde{\mu}}$ is of TYPE 1 and if G_μ is a path of length two, then $\tilde{\mu}$ admits all integer rectilinear spirality values in the range $[-2, 2]$, whereas μ does not admit any RU-spirality whose corresponding rectilinear spirality is -2 or 2. The next lemma shows that we can perform the RU-planarity testing of G_ν by looking at $\tilde{G_\nu}$, while avoiding for the gadgets of the children of ν in B those RU-spirality values that cannot be taken by the corresponding original components.

Lemma 17. *Let ν be an R-node and let $n_{\mathrm{skel}(\nu)}$ be the number of vertices in* skel(ν)*. Let $\mu_1, \ldots, \mu_a$ be the children of ν that belong to the set A. For each $i \in \{1, \ldots, a\}$, let τ_i be any given RU-spirality in the feasible set of μ_i. Also, let τ_ν be a target RU-spirality for ν. It is possible to test in $O(2^{2k} n_{\mathrm{skel}(\nu)})$ time whether G_ν admits an RU representation H_ν with RU-spirality τ_ν, such that the restriction of H_ν to G_{μ_i} has spirality τ_i, for each $i \in \{1, \ldots, a\}$.*

Proof. Consider the plane digraph $\tilde{G}_\nu$ obtained from skel(ν) by replacing each virtual edge with its corresponding gadget. For each of the $O(3^k) = O(2^{2k})$ upward planar embeddings of $\tilde{G}_\nu$, we test the existence of an RU-representation $\tilde{H}_\nu$ of $\tilde{G}_\nu$ by a 2-SAT formula Φ that enriches the one described in Theorem 3 with a suitable set of clauses.

More precisely, for each $\mu_i \in A$ denote by H_{μ_i} a representation of G_{μ_i} with the desired RU-spirality τ_i; we add to Φ a set of clauses that fix for the edges of G_{μ_i} the directions they have in H_{μ_i}. Concerning any element $\mu \in B$, recall that by Lemma 16 all RU-spirality values admitted by μ are also admitted by $\tilde{\mu}$. Also, all the planar embeddings of $G_{\tilde{\mu}}$ around its poles are equivalent (namely, $G_{\tilde{\mu}}$ has only one embedding if it is of TYPE 1, whereas it has two fully symmetric embeddings if it is of TYPE 2 or TYPE 3), hence it is not restrictive that $G_{\tilde{\mu}}$ has a fixed planar embedding in $\tilde{G_\nu}$. However, we must add to Φ clauses that prevent an RU representation of $G_{\tilde{\mu}}$ from taking an RU-spirality value that is not in the spirality set Σ_μ.

Formally, let $\tilde{\tau} = \langle \tilde{\sigma}, \varphi_u, \varphi_v \rangle \in \Sigma_{\tilde{\mu}} \setminus \Sigma_\mu$. In the following, we describe the clauses that must be added to Φ to prevent $G_{\tilde{\mu}}$ from taking RU-spirality $\tilde{\tau}$ in a representation $H_{\tilde{\mu}}$. We consider the case $\tilde{\sigma} \geq 0$; the case of $\tilde{\sigma} < 0$ can be handled in a symmetric way.

Suppose first that $G_{\tilde{\mu}}$ is TYPE 1 gadget, and refer to the notation of Fig. 19a. Let w_1, w_2, and w_3 be the vertices of $G_{\tilde{\mu}}$ distinct from u and v, so that $e_1 = (u, w_1)$, $e_2 = (w_1, w_2)$, and $e_3 = (w_2, v)$. We have that $\tilde{\sigma} \in \{0, 1, 2\}$ (see Fig. 20). Also, the poles u and v of G_μ are incident to exactly one internal edge of G_μ. Hence, for each $w \in \{u, v\}$, exactly one element in the pole specification φ_w equals int. By the same argument as in Lemma 8, since e_2 is the only outgoing edge of w_1 and the only incoming edge of w_2, we can assume that $b^U_{e_2}$ is set to `True`.

- `Case` $\tilde{\sigma} = 0$. We have one of the following situations for the pole specifications of $\tilde{\tau}$: (1) $N_u = S_v = \text{int}$ (see Fig. 20a); (2) $E_u = W_v = \text{int}$ (see Fig. 20b); (3) $W_u = E_v = \text{int}$ (see Fig. 20c). Depending on each of these situations, we add one of the following clauses to avoid that $\tilde{\sigma}$ is taken by an RU-representation of $G_{\tilde{\mu}}$: (1) $cl(\tilde{\tau}) = (\neg b^U_{e_1} \vee \neg b^U_{e_3})$; (2) $cl(\tilde{\tau}) = (\neg b^R_{e_1} \vee \neg b^R_{e_3})$; (3) $cl(\tilde{\tau}) = (\neg b^L_{e_1} \vee \neg b^L_{e_3})$.
- `Case` $\tilde{\sigma} = 1$. We have one of the following situations for the pole specifications of $\tilde{\tau}$: (1) $N_u = W_v = \text{int}$ (see Fig. 20d); (2) $W_u = S_v = \text{int}$ (see Fig. 20e). Again, for each of these situations, we exclude $\tilde{\tau}$ by adding the following clauses: (1) $cl(\tilde{\tau}) = (\neg b^U_{e_1} \vee \neg b^R_{e_3})$; (2) $cl(\tilde{\tau}) = (\neg b^L_{e_1} \vee \neg b^U_{e_3})$.
- `Case` $\tilde{\sigma} = 2$. We have $W_u = W_v = \text{int}$ (Fig. 20f). To exclude $\tilde{\tau}$ we add the clause $cl(\tilde{\tau}) = (\neg b^L_{e_1} \vee \neg b^R_{e_3})$.

Suppose now that $G_{\tilde{\mu}}$ is a TYPE 2 gadget, and refer to the notation of Fig. 19b. We have $\tilde{\sigma} \in \{\frac{1}{2}, \frac{3}{2}\}$. We analyze the case in which v has two alias vertices v' and v''; the case in which u has two alias vertices is symmetric. We have that either e_3 is directed towards v'' or it is directed towards v. We analyze the former case; the latter case can be handled with similar considerations.

- `Case` $\tilde{\sigma} = \frac{1}{2}$. We have $N_u = \text{int}$ and $S_v = W_v = \text{int}$ (see Fig. 21a). Observe that since v is a degree-4 vertex, $\neg b^R_{e_3}$ implies $\neg b^U_{e_2}$ by the clauses added initially (see Theorem 3). Hence: $cl(\tilde{\tau}) = (\neg b^R_{e_3} \vee \neg b^U_{e_1})$.
- `Case` $\tilde{\sigma} = \frac{3}{2}$. $S_v = W_v = \text{int}$, and $W_u = \text{int}$, see Fig. 21b. Again, since v is a degree-4 vertex, $\neg b^R_{e_3}$ implies $\neg b^U_{e_2}$. Hence, if: $cl(\tilde{\tau}) = (\neg b^R_{e_3} \vee \neg b^L_{e_1})$.

Suppose that G_μ and $G_{\tilde{\mu}}$ are of TYPE 3. We use the notation of Fig. 19c. As in the previous case, by Theorem 3, since v is a degree-4 vertex, $\neg b^R_{e_4}$ or $\neg b^L_{e_4}$ implies $\neg b^U_{e_3}$ and vice versa. The same properties holds for u with respect to edges e_1 and e_2. We assume e_1 and e_2 directed towards u, and e_3 and e_4 directed towards v' and v'', respectively. The other cases can be handled similarly.

- `Case` $\tilde{\sigma} = 0$. In this case we have one of the following situations: (*a*) $N_u = W_u = S_v = E_v = \text{int}$, see Fig. 22a; (*b*) $N_u = E_u = S_v = W_v = \text{int}$, see Fig. 22b. Depending on which case we are, we add one the following clauses: (*a*) $cl(\tilde{\tau}) = (\neg b^U_{e_1} \vee \neg b^U_{e_4})$; (*b*) $cl(\tilde{\tau}) = (\neg b^U_{e_3} \vee \neg b^U_{e_2})$.
- `Case` $\tilde{\sigma} = 1$. $S_v = W_v = \text{int}$, and $N_u = W_u = \text{int}$, see Fig. 22c. Hence: $cl(\tilde{\tau}) = (\neg b^U_{e_1} \vee \neg b^U_{e_3})$.

We add the clauses according to the case analysis above for each $\mu \in B$ and for each $\tilde{\tau} \in \Sigma_{\tilde{\mu}} \setminus \Sigma_\mu$. If the test is positive, we can simply construct H_ν from $\tilde{H_\nu}$ by selecting, for each child μ of ν corresponding to a virtual edge of skel(ν), a representation in the spirality set of μ having the same RU-spirality as its corresponding gadget. □

Lemma 18. *Let ν be an R-node with children $\mu_1, \ldots, \mu_h$, and let $n_{\text{skel}(\nu)}$ be the number of vertices in* skel(ν)*. Given the feasible set Σ_{μ_i} for each $i \in \{1, \ldots, h\}$, the feasible set Σ_ν of ν can be computed in $O(2^{k \log k + 2k} \cdot n_{\text{skel}(\nu)})$ time.*

Proof. For each of the two planar embeddings of skel(ν), for each possible combination of RU-spirality values admitted by the child nodes of ν that belong to A, and for each target RU-spirality τ_ν of G_ν, we apply Lemma 17 to test in $O(2^{2^k} n_{\text{skel}(\nu)})$ time whether there exists an RU representation of G_ν with RU-spirality τ_ν which preserves the given RU-spirality values for the children of ν in A. Since $|A| = O(k)$ and since by Lemma 14 there are $O(k)$ target RU-spirality values for ν, the total time required to construct Σ_ν is $O(k^k 2^{2k} n_{\text{skel}(\nu)}) = O(2^k \log k 2^{2k} n_{\text{skel}(\nu)}) = O(2^{k \log k + 2k} n_{\text{skel}(\nu)})$. □

Theorem 6. *Let G be a biconnected planar digraph with n vertices and k switches. There exists an $O(2^{k \cdot \log k + 2k} \cdot n)$-time algorithm that tests whether G is rectilinear-upward planar and that computes an RU representation of G in the positive case.*

Proof. The algorithm extends the dynamic programming technique used for SP-digraphs. Namely, Q-, S- and P-nodes are processed as described in Section 6.2, but now, thanks to Lemma 14, the size of the spirality set of each node is

$O(k)$. Also, each RU representation of G must have at least two switches of G on its external face, thus it is enough to explore only those rooted SPQR-trees for which the reference edge represented by the root is incident to a switch of G. The number of these trees is $O(k)$. Hence, by Lemma 13, all the spirality sets of Q-, S-, and P-nodes are computed in $O(k^2)$ over all rooted SPQR-trees. Also, by Lemma 18, the computation of the spirality set of an R-node in each rooted tree takes $O(2^{k\cdot\log k+2k}\cdot n_{\mathrm{skel}(\nu)})$-time. Since $\sum_\nu n_{\mathrm{skel}(\nu)} = O(n)$, processing all R-nodes of a single rooted SPQR-tree takes $O(2^{k\cdot\log k+2k}\cdot n)$ time. Since we need to explore only $O(k)$ rooted SPQR-trees, the theorem follows. □

The following corollary is an immediate consequence of Theorem 6 when the digraph has only one source and only one sink, i.e., when it is a planar st-digraph.

Corollary 1. *The* Rectilinear-Upward Planarity Testing *problem can be solved in $O(n)$ time for biconnected planar st-digraphs with n vertices.*

7. Conclusions and open problems

In this paper we addressed rectilinear-upward planar drawings of digraphs, which combine two of the most studied conventions for graph visualization. In the variable embedding setting, we proved that testing whether a planar digraph admits a rectilinear-upward planar drawing is an NP-complete problem, even for biconnected digraphs. On the positive side, we proved that the problem is solvable in quadratic time for biconnected series-parallel graphs and it is fixed-parameter linear (FPL) for all biconnected digraphs. Finally, if the input digraph has a fixed upward planar embedding, the problem can be solved in linear time. We conclude by highlighting two future research directions that naturally arise from our results.

- The NP-hardness of Rectilinear-Upward Planarity Testing holds when the embedding can vary while the linear-time solution holds for upward plane digraphs. Also the testing is trivial if a rectilinear embedding is given. Establishing the complexity of the problem when a planar embedding (neither rectilinear nor upward) is fixed remains an open question.
- Our algorithms in the variable embedding setting apply to biconnected digraphs; establishing whether these algorithms can be extended to simply connected digraphs is an interesting open question.

CRediT authorship contribution statement

Walter Didimo: Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Michael Kaufmann:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Giuseppe Liotta:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Giacomo Ortali:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Maurizio Patrignani:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank Ignaz Rutter for conversations about the problem 1-2-Switch-Flow.

Data availability

No data was used for the research described in the article.

References

[1] G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis, Graph Drawing: Algorithms for the Visualization of Graphs, Prentice-Hall, 1999.
[2] T. Nishizeki, M.S. Rahman, Planar Graph Drawing, Lecture Notes Series on Computing, vol. 12, World Scientific, 2004.
[3] M. Patrignani, Planarity testing and embedding, in: R. Tamassia (Ed.), Handbook on Graph Drawing and Visualization, Chapman and Hall/CRC, 2013, pp. 1–42.
[4] W. Didimo, G. Liotta, Graph Visualization and Data Mining, John Wiley & Sons, Ltd, 2006, pp. 35–63, Ch. 3.
[5] M. Jünger, P. Mutzel (Eds.), Graph Drawing Software, Springer, 2004.
[6] A. Garg, R. Tamassia, On the computational complexity of upward and rectilinear planarity testing, SIAM J. Comput. 31 (2) (2001) 601–625, https://doi.org/10.1137/S0097539794277123.
[7] R. Tamassia, On embedding a graph in the grid with the minimum number of bends, SIAM J. Comput. 16 (3) (1987) 421–444, https://doi.org/10.1137/0216030.

[8] G. Di Battista, G. Liotta, F. Vargiu, Spirality and optimal orthogonal drawings, SIAM J. Comput. 27 (6) (1998) 1764–1811, https://doi.org/10.1137/S0097539794262847.
[9] E. Di Giacomo, G. Liotta, F. Montecchiani, Sketched representations and orthogonal planarity of bounded treewidth graphs, in: D. Archambault, C.D. Tóth (Eds.), Graph Drawing and Network Visualization - 27th International Symposium, GD 2019, in: Lecture Notes in Computer Science, vol. 11904, Springer, 2019, pp. 379–392.
[10] E. Di Giacomo, G. Liotta, F. Montecchiani, Orthogonal planarity testing of bounded treewidth graphs, J. Comput. Syst. Sci. 125 (2022) 129–148, https://doi.org/10.1016/j.jcss.2021.11.004.
[11] W. Didimo, G. Liotta, G. Ortali, M. Patrignani, Optimal orthogonal drawings of planar 3-graphs in linear time, in: S. Chawla (Ed.), Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, SIAM, 2020, pp. 806–825.
[12] F. Frati, Planar rectilinear drawings of outerplanar graphs in linear time, Comput. Geom. 103 (2022) 101854, https://doi.org/10.1016/j.comgeo.2021.101854.
[13] M.M. Hasan, M.S. Rahman, No-bend orthogonal drawings and no-bend orthogonally convex drawings of planar graphs (extended abstract), in: D. Du, Z. Duan, C. Tian (Eds.), Computing and Combinatorics - 25th International Conference, COCOON 2019, in: Lecture Notes in Computer Science, vol. 11653, Springer, 2019, pp. 254–265.
[14] M.S. Rahman, N. Egi, T. Nishizeki, No-bend orthogonal drawings of subdivisions of planar triconnected cubic graphs, IEICE Trans. Inf. Syst. 88-D (1) (2005) 23–30.
[15] M.S. Rahman, T. Nishizeki, M. Naznin, Orthogonal drawings of plane graphs without bends, J. Graph Algorithms Appl. 7 (4) (2003) 335–362, https://doi.org/10.7155/jgaa.00074.
[16] S. Abbasi, P. Healy, A. Rextin, Improving the running time of embedded upward planarity testing, Inf. Process. Lett. 110 (7) (2010) 274–278, https://doi.org/10.1016/j.ipl.2010.02.004.
[17] P. Bertolazzi, G. Di Battista, G. Liotta, C. Mannino, Upward drawings of triconnected digraphs, Algorithmica 12 (6) (1994) 476–497, https://doi.org/10.1007/BF01188716.
[18] P. Bertolazzi, G. Di Battista, C. Mannino, R. Tamassia, Optimal upward planarity testing of single-source digraphs, SIAM J. Comput. 27 (1) (1998) 132–169, https://doi.org/10.1137/S0097539794279626.
[19] C. Binucci, W. Didimo, M. Patrignani, Upward and quasi-upward planarity testing of embedded mixed graphs, Theor. Comput. Sci. 526 (2014) 75–89, https://doi.org/10.1016/j.tcs.2014.01.015.
[20] S. Chaplick, E. Di Giacomo, F. Frati, R. Ganian, C.N. Raftopoulou, K. Simonov, Parameterized algorithms for upward planarity, in: X. Goaoc, M. Kerber (Eds.), 38th International Symposium on Computational Geometry, SoCG 2022, in: LIPIcs, vol. 224, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, pp. 26:1–26:16.
[21] W. Didimo, F. Giordano, G. Liotta, Upward spirality and upward planarity testing, SIAM J. Discrete Math. 23 (4) (2009) 1842–1899, https://doi.org/10.1137/070696854.
[22] P. Healy, K. Lynch, Two fixed-parameter tractable algorithms for testing upward planarity, Int. J. Found. Comput. Sci. 17 (5) (2006) 1095–1114, https://doi.org/10.1142/S0129054106004285.
[23] M.D. Hutton, A. Lubiw, Upward planning of single-source acyclic digraphs, SIAM J. Comput. 25 (2) (1996) 291–311, https://doi.org/10.1137/S0097539792235906.
[24] F. Brandenburg, D. Eppstein, M.T. Goodrich, S.G. Kobourov, G. Liotta, P. Mutzel, Selected open problems in graph drawing, in: G. Liotta (Ed.), Graph Drawing, 11th International Symposium, GD 2003, in: Lecture Notes in Computer Science, vol. 2912, Springer, 2003, pp. 515–539.
[25] R. Tamassia, G. Liotta, Graph drawing, in: J.E. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, second edition, Chapman and Hall/CRC, 2004, pp. 1163–1185.
[26] S. Chaplick, E. Di Giacomo, F. Frati, R. Ganian, C.N. Raftopoulou, K. Simonov, Testing upward planarity of partial 2-trees, in: P. Angelini, R. von Hanxleden (Eds.), Graph Drawing and Network Visualization - 30th International Symposium, GD 2022, in: Lecture Notes in Computer Science, vol. 13764, Springer, 2022, pp. 175–187.
[27] W. Didimo, M. Kaufmann, G. Liotta, G. Ortali, Rectilinear planarity of partial 2-trees, in: P. Angelini, R. von Hanxleden (Eds.), Graph Drawing and Network Visualization - 30th International Symposium, GD 2022, in: Lecture Notes in Computer Science, vol. 13764, Springer, 2022, pp. 157–172.
[28] W. Didimo, M. Kaufmann, G. Liotta, G. Ortali, Rectilinear planarity of partial 2-trees, J. Graph Algorithms Appl. 27 (8) (2023) 679–719, https://doi.org/10.7155/JGAA.00640.
[29] E. Di Giacomo, W. Didimo, G. Liotta, F. Montecchiani, G. Ortali, On the parameterized complexity of bend-minimum orthogonal planarity, in: M.A. Bekos, M. Chimani (Eds.), Graph Drawing and Network Visualization - 31st International Symposium, GD 2023, in: Lecture Notes in Computer Science, vol. 14466, Springer, 2023, pp. 53–65.
[30] B.M.P. Jansen, L. Khazaliya, P. Kindermann, G. Liotta, F. Montecchiani, K. Simonov, Upward and orthogonal planarity are W[1]-hard parameterized by treewidth, in: M.A. Bekos, M. Chimani (Eds.), Graph Drawing and Network Visualization - 31st International Symposium, GD 2023, in: Lecture Notes in Computer Science, vol. 14466, Springer, 2023, pp. 203–217.
[31] S. Cornelsen, A. Karrenbauer, Accelerated bend minimization, J. Graph Algorithms Appl. 16 (3) (2012) 635–650, https://doi.org/10.7155/jgaa.00265.
[32] A. Garg, R. Tamassia, A new minimum cost flow algorithm with applications to graph drawing, in: S.C. North (Ed.), Graph Drawing, Symposium on Graph Drawing, GD '96, in: Lecture Notes in Computer Science, vol. 1190, Springer, 1996, pp. 201–216.
[33] W. Didimo, G. Liotta, M. Patrignani, HV-planarity: algorithms and complexity, J. Comput. Syst. Sci. 99 (2019) 72–90, https://doi.org/10.1016/j.jcss.2018.08.003.
[34] J. Hopcroft, R.E. Tarjan, Efficient planarity testing, J. ACM 21 (4) (1974) 549–568.
[35] R. Tamassia, I.G. Tollis, Planar grid embedding in linear time, IEEE Trans. Circuits Syst. CAS-36 (9) (1989) 1230–1234.
[36] G. Vijayan, A. Wigderson, Rectilinear graphs and their embeddings, SIAM J. Comput. 14 (1985) 355–372.
[37] M.P. Bonacina, N. Dershowitz, Canonical inference for implicational systems, in: A. Armando, P. Baumgartner, G. Dowek (Eds.), Automated Reasoning, 4th International Joint Conference, IJCAR 2008, in: Lecture Notes in Computer Science, vol. 5195, Springer, 2008, pp. 380–395.
[38] U. Fößmeier, M. Kaufmann, On bend-minimum orthogonal upward drawing of directed planar graphs, in: R. Tamassia, I.G. Tollis (Eds.), Graph Drawing, DIMACS International Workshop, GD '94, in: Lecture Notes in Computer Science, vol. 894, Springer, 1994, pp. 52–63.
[39] U. Fossmeier, M. Kaufmann, An approach to bend-minimal upward drawing, in: Graph Drawing, 1th International Symposium, GD 1993, Paris, France, 1993, pp. 27–29.
[40] B. Aspvall, M.F. Plass, R.E. Tarjan, A linear-time algorithm for testing the truth of certain quantified Boolean formulas, Inf. Process. Lett. 8 (3) (1979) 121–123, https://doi.org/10.1016/0020-0190(79)90002-4.
[41] S. Even, A. Itai, A. Shamir, On the complexity of timetable and multicommodity flow problems, SIAM J. Comput. 5 (4) (1976) 691–703, https://doi.org/10.1137/0205048.
[42] G. Di Battista, R. Tamassia, Algorithms for plane representations of acyclic digraphs, Theor. Comput. Sci. 61 (1988) 175–198, https://doi.org/10.1016/0304-3975(88)90123-5.
[43] J.E. Hopcroft, R.E. Tarjan, Dividing a graph into triconnected components, SIAM J. Comput. 2 (3) (1973) 135–158.
[44] C. Gutwenger, P. Mutzel, A linear time implementation of SPQR-trees, in: J. Marks (Ed.), Graph Drawing, 8th International Symposium, GD 2000, in: Lecture Notes in Computer Science, vol. 1984, Springer, 2000, pp. 77–90.

[45] W. Didimo, M. Kaufmann, G. Liotta, G. Ortali, Computing bend-minimum orthogonal drawings of plane series–parallel graphs in linear time, Algorithmica (2023), https://doi.org/10.1007/s00453-023-01110-6.
[46] W. Didimo, G. Liotta, Computing orthogonal drawings in a variable embedding setting, in: K. Chwa, O.H. Ibarra (Eds.), Algorithms and Computation, 9th International Symposium, ISAAC '98, in: Lecture Notes in Computer Science, vol. 1533, Springer, 1998, pp. 79–88.
[47] G. Di Battista, R. Tamassia, On-line planarity testing, SIAM J. Comput. 25 (5) (1996) 956–997, https://doi.org/10.1137/S0097539794280736.
[48] S. Even, R.E. Tarjan, Corrigendum: computing an st-numbering. TCS 2(1976):339–344, Theor. Comput. Sci. 4 (1) (1977) 123.