The st-Planar Edge Completion Problem is Fixed-Parameter Tractable

Liana Khazaliya
Technische Universität Wien, Austria

Philipp Kindermann
FB IV - Informatikwissenschaften, Universität Trier, Germany

Giuseppe Liotta
Department of Engineering, University of Perugia, Italy

Fabrizio Montecchiani
Department of Engineering, University of Perugia, Italy

Kirill Simonov
Hasso Plattner Institute, University of Potsdam, Germany

Abstract

The problem of deciding whether a biconnected planar digraph $G = (V, E)$ can be augmented to become an st-planar graph by adding a set of oriented edges $E' \subseteq V \times V$ is known to be NP-complete. We show that the problem is fixed-parameter tractable when parameterized by the size of the set E'.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability; Mathematics of computing → Graph algorithms

Keywords and phrases st-planar graphs, parameterized complexity, upward planarity

Funding Research of GL and FM partially supported by MUR of Italy, under PRIN Project n. 2022ME9Z78 - NextGRAAL: Next-generation algorithms for constrained GRAph visuALization, and under PRIN Project n. 2022TS4Y3N - EXPAND: scalable algorithms for EXPloratory Analyses of heterogeneous and dynamic Networked Data.

Acknowledgements This research was initiated at Dagstuhl Seminar 23162: New Frontiers of Parameterized Complexity in Graph Drawing.

1 Introduction

Edge modification problems have long been a subject of investigation in graph algorithms, resulting in a vast body of literature dedicated to exploring their computational complexity (refer, for instance, to Burzyn et al. [4] and to Natanzon et al. [16] for comprehensive surveys). One specific category within this realm is the family of edge completion problems, which can be succinctly described as follows: Given a graph $G = (V, E)$ and a graph family $\mathcal{G}$, the objective is to determine whether it is possible to augment G with a set $E' \subseteq V \times V$ of edges such that $G' = (V, E \cup E') \in \mathcal{G}$. In such cases, we say that G becomes a member of $\mathcal{G}$ by *adding* the edges in E'. Edge completion problems are frequently known to be NP-hard, thereby inspiring numerous studies focusing on parameterized complexity. For a comprehensive examination of parameterized algorithms addressing edge completion problems, we point the reader to the exhaustive survey by Crespelle et al. [7].

This paper focuses on the investigation of an edge completion problem specifically applied to directed graphs (*digraphs* for short). More precisely, let $G = (V, E)$ be a digraph. A vertex of G with no incoming edges is a *source* of G, while a vertex without outgoing edges is a *sink* of G. A digraph G is an *st-planar graph* if it admits a planar embedding such that: (1) it contains no directed cycle; (2) it contains a single source vertex s and a single sink vertex t; (3) s and t both belong to the external face of the planar embedding.

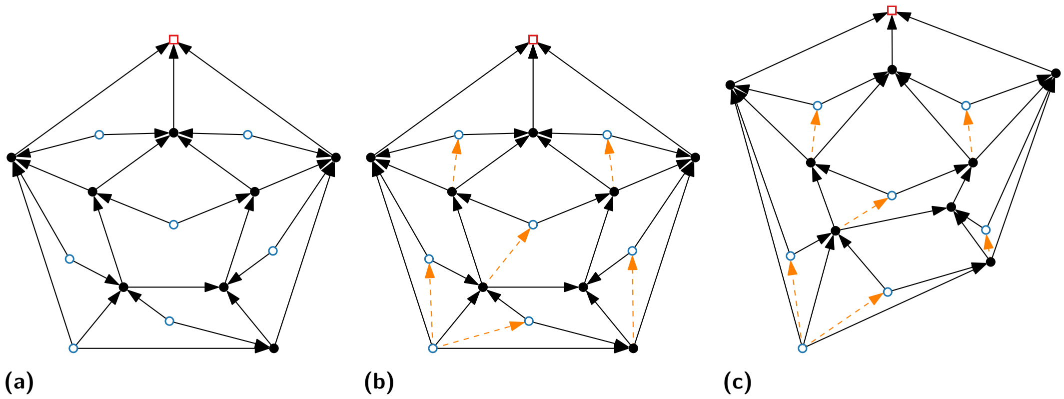


Figure 1 (a) A digraph G with $2k+1=7$ sources and 1 sink; G has a unique planar embedding up to the choice of the external face; (b) A completion of G to an st-planar graph obtained by adding $2k=6$ edges; (c) An upward planar drawing of the completion of G.

Upward planarity is a rather natural and well-studied notion of planarity for directed graphs (see, e.g., [5, 6, 8, 10, 13, 17]). In particular, a planar digraph is *upward* if it admits a planar drawing where all edges are oriented upward. A well-known result in graph drawing states that a digraph G is upward if and only if G is a subgraph of an st-planar graph [8, 10][1]. However, since testing for upward planarity is an NP-complete problem already for biconnected graphs [13], determining whether a biconnected graph is a subgraph of an st-planar graph is also computationally challenging. On the other hand, checking whether a digraph is st-planar can be done efficiently in polynomial time. This observation motivates for the investigation of the following problem.

st-Planar Edge Completion (st-PEC)
Input: A biconnected digraph G
Parameter: $k \in \mathbb{N}$
Question: Is it possible to add at most k edges to G such that the resulting graph is an st-planar graph?

In this paper, we present a fixed-parameter tractable algorithm for the st-Planar Edge Completion problem. To help understanding the combinatorial and algorithmic challenges behind the problem, we make the observation that the parameter k provides an upper bound on the number of sources and sinks in the input digraph G. Since an edge can remove the presence of at most one source and one sink, if the total number of sources and sinks in G exceeds $2k+2$, we can promptly reject the instance. Conversely, a positive answer to st-Planar Edge Completion implies that G is upward planar. In this respect, it is worth mentioning that Chaplick et al. [5] have previously demonstrated that testing a digraph for upward planarity is fixed-parameter tractable when parameterized by the number of its sources. However, for every $k \geq 1$, there are upward planar digraphs with at most $2k+1$ sources that cannot be augmented to an st-planar graph by adding k edges; refer to Figure 1 for an illustration. Furthermore, while an upward planarity test halts upon finding an upward planar embedding, not all upward planar embeddings of the same digraph can lead to an st-planar graph after the addition of k edges. Figure 2 demonstrates an upward planar

[1] From the proof in Lemma 4.1 of [10], one can in fact observe that a digraph is upward planar if and only if it is a subgraph of an st-planar graph defined over the same set of vertices.

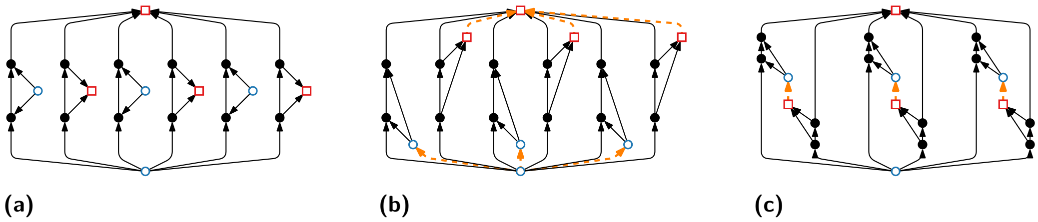


Figure 2 (a) A biconnected digraph G with 4 sources and 4 sinks; (b) With the given embedding, 6 edges have to be added to complete G to an st-planar graph; (c) With a different embedding, adding 3 edges is sufficient.

digraph along with two of its upward planar embeddings: the embedding in Figure 2a requires 6 edges to be augmented into an st-planar digraph, whereas the embedding in Figure 2c can be augmented with 3 edges.

In order to overcome the above technical challenges, our result is based on a structural decomposition of the digraph into its triconnected components using SPQR-trees (similarly as done in [5]), as well as on novel insights regarding the combinatorial properties of upward planar digraphs. Since the proof is rather technical, after giving preliminaries and basic notation in Section 2, we present an overview of the approach in Section 3. Next, the FPT algorithm is described in full detail in Section 4. We conclude in Section 5.

2 Preliminaries

In this section, we provide basic definitions and tools that will be used throughout the paper.

Planar drawings and embeddings. A *planar drawing* of a graph G maps the vertices of G to points of the plane and the edges of G to Jordan arcs such that no two arcs share a point except at common end-vertices. A planar drawing partitions the plane into topologically connected regions called *faces*, one of which is unbounded and called the *external face*, in contrast with all other faces which are *inner faces*. For a digraph G, a planar drawing is called *upward* if each edge oriented from a vertex u to a vertex v is represented by a Jordan arc monotonically increasing from the point representing u to the point representing v. A graph (digraph) is *planar* (*upward planar*) if it admits a planar drawing (upward planar drawing). A *planar embedding* (*upward planar embedding*) $\mathcal{E}(G)$ of a planar graph (upward planar digraph) G represents an equivalence class of planar drawings (upward planar drawings) with the same inner faces and the same external face, up to a homeomorphism of the plane. Graph G is *plane* if it comes with a fixed planar embedding $\mathcal{E}(G)$.

SPQR-trees. We recall the definition of *SPQR-tree*, introduced in [8], which represents the decomposition of a biconnected graph G into its triconnected components [15]. Each triconnected component corresponds to a non-leaf node ν of T; the triconnected component itself is called the *skeleton* of ν and is denoted as skel(ν). Node ν can be: (*i*) an *R-node*, if skel(ν) is a triconnected graph; (*ii*) an *S-node*, if skel(ν) is a simple cycle of length at least three; (*iii*) a *P-node*, if skel(ν) is a bundle of at least three parallel edges. A degree-1 node of T is a *Q-node* and represents a single edge of G. A *real edge* (resp. *virtual edge*) in skel(ν) 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 [8, 14]. 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, skel(ν) has a virtual edge, called *reference edge* of ν and denoted as e_ν, associated with a virtual edge in the skeleton of its parent. The end-vertices of the reference edge of ν are called the *poles* of ν. 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. Without loss of generality, we shall consider SQPR-trees where every S-node has exactly two children (see, e.g., [5, 9, 12]); this lifts the condition that two S-nodes cannot be adjacent in T.

Angles in upward drawings. Let $G = (V, E)$ be a digraph. For each edge $(u, v) \in E$, we write uv if (u, v) is oriented from u to v in G, and we write vu otherwise. A vertex v is a *switch* of G, if it is either a source or a sink, and it is a *non-switch* otherwise. Recall that a digraph is upward planar if and only if it is a subgraph of an st-planar graph [8]. Hence, being upward planar is a necessary condition for YES-instances of st-Planar Edge Completion. Consider now a biconnected plane digraph G. An *angle* is an incidence between a vertex v and a face f of G. Let α be one such angle, and consider the two edges incident to v that belong to the boundary of f. If such edges are one incoming and one outgoing, α is a *non-switch angle*, while if the edges are both incoming or both outgoing, α is a *switch angle*. Note that a switch angle in a face f can be made by two edges that are incident to a non-switch vertex v: it is enough that the edges of f incident to v are both incoming or both outgoing. In this case, v is a *local switch* for face f. An *angle assignment* is a labeling λ of the angles of G with labels $\{-1, 0, +1\}$ (see, e.g., [1, 2, 3, 11]). In particular, non-switch angles can only receive the label 0, while switch angles can be labeled as either -1 or $+1$. The planar embedding of G can be realized as an upward drawing if and only if there is an angle assignment such that: (i) each switch vertex has exactly one angle labeled $+1$; (ii) each non-switch vertex has exactly two angles labeled as 0, while all the others are switch angles labeled -1; (iii) the difference between the number of angles labeled $+1$ and the number of angles labeled -1 along the boundary of each inner face is -2; (iv) the difference between the number of angles labeled $+1$ and the number of angles labeled -1 along the boundary of the external face is $+2$. Observe that property (ii) implies that each non-switch vertex forms exactly two non-switch angles. An angle assignment satisfying the above properties is called *upward*. The restriction of an upward angle assignment to the angles of a single face f is an *upward angle assignment for f*.

3 Overview of the Approach

Let G be a biconnected digraph. Since testing for planarity can be done in linear time, we shall assume that G is planar. We begin by explaining two key ingredients for our algorithm, namely, the use of SPQR-trees to encode all the planar embeddings of G, and the use of upward angle assignments to incrementally saturate G. The main crux of our algorithm lies in blending these two ingredients together to design a dynamic program that solves the problem in FPT time.

Let T be a rooted SPQR-tree of a planar graph G with reference edge e. The planar embeddings of G in which the edge e lies on the boundary of the external face can be obtained as follows (see, e.g., [8]). 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. No operations are needed at S- and Q-nodes.

Consider now an upward planar drawing Γ of G and hence assume that G is plane. Let λ be the upward angle assignment *induced* by Γ. Precisely, the switch angles that are larger (smaller) than π in Γ are labeled as $+1$ (-1), while the non-switch angles are labeled as 0. Let v be a source (sink) of G and let f be the face of G in which v makes its $+1$ angle. Let

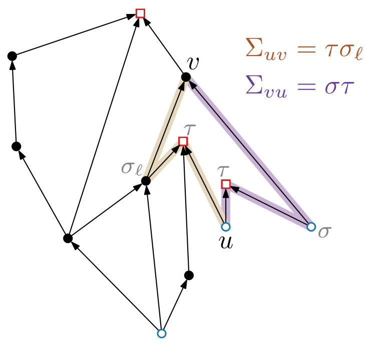


Figure 3 The signatures of two paths Π_{uv} (brown background) and Π_{vu} (purple background).

u be a vertex of f different from v. We say that adding uv (vu) to G *saturates* v, and that uv (vu) is a *saturating edge*. Namely, v becomes a non-switch vertex in $G' = (V, E \cup \{uv\})$. Notably, f is the only face in which an edge saturating v can be added: one easily verifies that choosing any other face would lead to a non-upward angle assignment.

Based on the previous reasoning, at high-level, the algorithm will exploit a bottom-up traversal of the SPQR-tree T to explore the planar embeddings of G. For each visited node, it will keep track of the information related to the minimum number of edges required to saturate all switches that lie in the inner faces of the corresponding pertinent graph. The interface of a candidate solution is encoded in terms of “signatures” which, informally, are strings containing all switches along the boundary of the external face of the pertinent graph that do not yet have any angle labeled as $+1$ and all vertices that must instead contribute with a -1 angle along the boundary. A running time bounded by a function of the budget k is obtained by several crucial insights about how a bounded number of switches in the graph affects the possible signatures and limits the relevant embeddings to be considered.

4 An FPT Algorithm for st-Planar Edge Completion

In this section, we describe our FPT algorithm, which leads to the following theorem.

▶ **Theorem 1.** *Let G be an n-vertex biconnected plane digraph. There is an algorithm that solves st-Planar Edge Completion in $2^{O(k^2)} \cdot n^2$ time.*

We begin by describing the records used by our dynamic program (Section 4.1), which are used to encode the angles along the boundary of the external face of a pertinent graph. Next, we describe the algorithm (Section 4.2), which constructs such records while traversing bottom-up the SPQR-tree of the input graph.

4.1 Setting up the Records for Dynamic Programming

Signatures. We begin with some notation and definitions. Let G be a plane digraph. Let Π_{uv} be a simple undirected path of G from a vertex u to a vertex v. The *signature* of Π_{uv} is a string Σ_{uv} computed as follows. Consider a walk along Π_{uv} from u to v. For each encountered vertex w distinct from u and v, look at the two edges incident to w in Π_{uv}. If the two edges are one incoming and one outgoing, we do not append any symbol to Σ_{uv}. If the two edges are both outgoing (incoming) and w is a switch of G, we append the symbol σ (τ). If the two edges are both outgoing (incoming) and w is not a switch of G – hence, it is a local switch for some face f –, we append the symbol σ_ℓ (τ_ℓ). Observe that, if Π_{uv} is a single edge connecting u to v, then $\Sigma_{uv} = \emptyset$. At high level, the idea is that when walking along a piece of the boundary of some face f of G, non-switch angles are ignored as their only possible value in an angle assignment is 0. On the other hand, the symbols σ_ℓ and τ_ℓ will encode switch-angles whose only possible value is -1 (else the corresponding vertex would be a switch of G). Finally, the symbols σ and τ will point to switch angles that may be assigned either -1 or $+1$. Refer to Figure 3 for an illustration.

A signature is *short* if it contains at most $4k + 2$ symbols. Let Σ^* be the set of short signatures; we observe the following.

▶ **Observation 2.** *The cardinality of Σ^* is $2^{O(k)}$.*

Half-boundaries. Let G be a biconnected planar digraph, and let T be the SPQR-tree of G rooted at a Q-node representing an arbitrary edge e of G. For each node ν of T, we recall that G_ν is the pertinent graph, and we denote by u, v the poles of ν (omitting the dependency on ν for simplicity). Assuming that G_ν comes with a fixed planar embedding, let f be the external face of G_ν. The *half-boundary* B_{uv} of ν is the path containing the vertices of f encountered in a clockwise walk of the face from u to v. The half-boundary B_{vu} of ν is defined analogously walking from v to u. A vertex w on the boundary of f is *bifacial* if it belongs to both B_{uv} and B_{vu} (which implies that w is a cutvertex of G_ν and hence ν is an S-node). For each of the two half-boundaries we can define the two corresponding signatures Σ_{uv} and Σ_{vu}. We will assume that for each symbol of Σ_{uv} and Σ_{vu} we have a pointer to the corresponding vertex. Let B be one of the two half-boundaries of ν and let Σ be its signature. Let B' be a path contained in B (possibly $B = B'$). The *restriction* of Σ to B', denoted as $\Sigma[B']$, is the substring of Σ containing the symbols whose corresponding vertices belong to B'. The next lemma shows that working with short signatures is not restrictive.

▶ **Lemma 3.** *Let G be a biconnected upward planar digraph with a fixed upward planar embedding $\mathcal{E}(G)$. Let T be the SPQR-tree of G rooted at a Q-node representing an arbitrary edge e of G. Let ν be a node of T. For any fixed k, if G can be augmented to an st-planar graph by adding at most k saturating edges, then the signatures Σ_{uv} and Σ_{vu} of the two half-boundaries B_{uv} and B_{vu} of ν are both short.*

Proof. Let Γ be an upward planar drawing of G whose corresponding upward planar embedding is $\mathcal{E}(G)$, and consider the subdrawing Γ' induced by G_ν. Let λ be the upward angle assignment induced by Γ', and let f be the external face of G_ν. We know that f contains at most $2k + 2$ switches, otherwise k saturating edges would not suffice to turn G into an st-planar graph. Hence, λ can label $+1$ at most $2k + 2$ angles along the boundary of f. Also, since λ obeys to property (iv) of an upward angle assignment, it labels -1 at most $2k$ angles. Therefore, Σ_{uv} and Σ_{vu} can each contain at most $4k + 2$ symbols. ◀

Internal assignments. An angle of G_ν is *internal* if it is defined in an inner face of G_ν. An *internal assignment* of G_ν is an angle assignment λ that labels all the internal angles of G_ν and that respects properties (i)–(iii) for upward angle assignments (but ignoring property (iv)). A switch vertex of G is called *active* with respect to λ if none if its internal angles (if any) received value $+1$. The *cost* of an internal assignment λ of G_ν is the minimum number of saturating edges needed to saturate all switches of G_ν that are not active with respect to λ.

Partial solutions. We are now ready to define the table used by our dynamic program. A tuple $\langle \Sigma_1, \Sigma_2, b_1, b_2 \rangle$, such that Σ_1, Σ_2 is a pair of short signatures and b_1, b_2 is a pair of flags, is called a *candidate tuple* in the following. Given a node ν and a candidate tuple $\langle \Sigma_1, \Sigma_2, b_1, b_2 \rangle$, the function $X(\nu, \Sigma_1, \Sigma_2, b_u, b_v)$ returns the minimum cost of an internal assignment λ of G_ν such that: (1) Σ_1 and Σ_2 are the signatures of its two half-boundaries B_{uv} and B_{vu}, respectively, (2) the flag b_u is true if and only if u is an active switch with respect to λ, (3) the flag b_v is true if and only if v is an active switch with respect to λ. The set of *partial solutions* for ν is given by the restriction of X to the single node ν. Also, a pair of signatures is *empty* if both its signatures are empty (i.e., they do not contain any symbol).

4.2 Description of the Algorithm

The function X is computed by traversing T bottom-up. For each node ν of T, we initialize $X(\nu, \Sigma_1, \Sigma_2, b_1, b_2) = +\infty$ for each candidate tuple $\langle \Sigma_1, \Sigma_2, b_1, b_2 \rangle$. We only ensure that $X(\nu, \Sigma_1, \Sigma_2, b_1, b_2)$ is computed precisely if the value is at most k; for any value larger than k we assume that $X(\nu, \Sigma_1, \Sigma_2, b_1, b_2) = +\infty$ is the correct setting, since we are only interested in the solutions that add at most k edges.

If ν is a leaf node, then it is a Q-node and G_ν is a single edge. In this case, either u is the source and v is the sink of G_ν, or vice-versa. Then we set $X(\nu, \emptyset, \emptyset, \mathrm{true}, \mathrm{true}) = 0$.

The lemma below deals with the case in which ν is an S-node. Since S-nodes have exactly two children and are not used to describe the planar embeddings of G, the routine of the algorithm at S-nodes is relatively simple. Next, we will consider P-nodes and R-nodes, which require more involved arguments.

▶ **Lemma 4.** *Let ν be an S-node of T. The set of partial solutions of ν can computed in $2^{O(k)}$ time.*

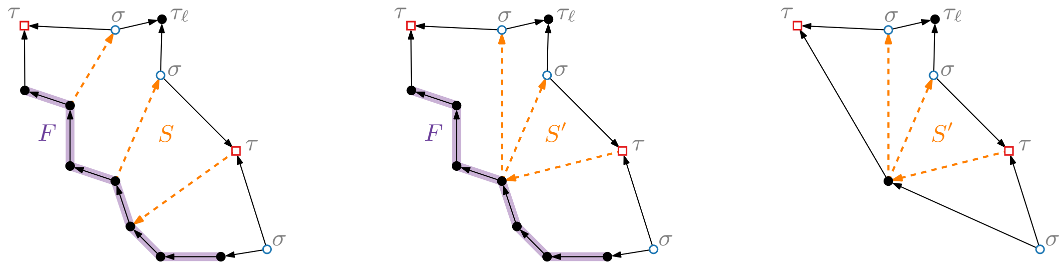
Proof. Let μ_1 and μ_2 be the two children of ν. In order to compute the partial solutions for ν, we check whether pairs of internal assignments of G_{μ_1} and G_{μ_2} can be combined together. Let $\langle \Sigma_{1,1}, \Sigma_{1,2}, b_{1,1}, b_{1,2} \rangle$ and $\langle \Sigma_{2,1}, \Sigma_{2,2}, b_{2,1}, b_{2,2} \rangle$ be a pair of candidate tuples. Also, let $C = X(\mu_1, \Sigma_{1,1}, \Sigma_{1,2}, b_{1,1}, b_{1,2}) + X(\mu_2, \Sigma_{2,1}, \Sigma_{2,2}, b_{2,1}, b_{2,2})$.

We first verify that $C \leq k$, and that $b_{1,2} \vee b_{2,1} = \mathrm{true}$. The first condition guarantees that we have not exceeded our budget k of saturating edges, while the second condition guarantees that the pole shared by μ_1 and μ_2 does not receive the value $+1$ twice in the final internal assignment of G_ν. If both conditions are satisfied, then we proceed as detailed below, otherwise, we discard the pair of candidate tuples.

Denote by u and w the poles of μ_1, and by w and v the poles of μ_2. Observe that B_{uv} corresponds to the union of B_{uw} and B_{wv} (vertex w is hence bifacial). Based on this observation, we show how to compute Σ_1 for B_{uv}, the computation of Σ_2 can be performed analogously. We initially set $\Sigma_1 = \Sigma_{1,1}$. Consider the two edges incident to w along B_{uv}. If one edge is incoming and the other is outgoing, then we do not append any symbol. If both edges are incoming or both outgoing, we check whether one of $b_{1,2}$ and $b_{2,1}$ is false. If so, we append the symbol σ_ℓ if w is a source of G, or the symbol τ_ℓ otherwise. If none of $b_{1,2}$ and $b_{2,1}$ is false, we append the symbol σ if w is a source of G, or the symbol τ otherwise. Next, we concatenate the signature $\Sigma_{2,1}$. Once both Σ_1 and Σ_2 have been computed, we verify that each of them is short (a necessary condition by Lemma 3), otherwise we discard the candidate tuples. Finally, we set $X(\nu, \Sigma_1, \Sigma_2, b_{1,1}, b_{2,2}) = C$.

By Observation 2, we have $2^{O(k)}$ possible pairs of signatures to consider, and performing the above operations takes $O(k)$ time for each pair. ◀

The next tools will be useful for the remaining lemmas. The next result is based on the fact that face boundaries containing long sequences of non-switch vertices are irrelevant for the sake of computing the least number of saturating edges; see Figure 4 for an illustration.



(a) **(b)** **(c)**

Figure 4 Illustration for Lemma 5.

▶ **Lemma 5.** *Let f be an inner face of G with n_f vertices, and let λ_f be an upward angle assignment for f with h switch-angles. The minimum number of edges that saturate all switch vertices of G forming an angle labeled $+1$ in f can be computed in $O(2^{O(h^2)} + n_f)$ time.*

Proof. Consider a sequence of non-switch angles in a walk along the boundary of f, and let F be the corresponding set of vertices. Also, consider a set of saturating edges S drawn inside f, such that each of them has one end-vertex that belongs to F; see Figure 5a. Since in any upward drawing of G the set F is drawn as a monotonically increasing curve, it is immediate to see that the set S can be replaced with a set of saturating edges S' such that: S and S' have the same size, S and S' saturate the same set of switches, all edges of S' are incident to a single (arbitrarily chosen) vertex of F; see Figure 5b. Consequently, we can work with a simplified boundary of f in which maximal sequences of vertices forming non-switch angles are replaced with a single vertex; see Figure 5c. Computing the simplified boundary takes $O(n_f)$ time, and such a boundary has at most $2h+1$ vertices. Thus, the maximum number of edges whose end-vertices belong to this boundary is $\binom{2h+1}{2}$, and in $2^{O(h^2)}$ time we can enumerate all candidate sets of saturating edges whose end-vertices belong to this boundary. Finally, among these sets, we return the size of the smallest set that saturates all switches labeled $+1$ and such that no two of its edges cross. (Whether two edges cross only depends on the order of their end-vertices along the boundary of f.) ◀

▶ **Lemma 6.** *Let ν be a node of T and let μ be a child of ν. Suppose that G_ν is plane and a half-boundary B of ν contains a half-boundary B' of μ (B and B' may possibly coincide). Given an internal assignment λ of G_ν and the signature of B', the restriction of the signature of B to B' can be computed in $O(k)$ time.*

Proof. Let Σ' be the signature of B', we compute the desired signature Σ as follows. If Σ' does not contain any symbol in $\{\sigma, \tau\}$ whose corresponding vertex is bifacial, then $\Sigma = \Sigma'$. Otherwise we initialize $\Sigma = \Sigma'$ and proceed as follows. For each σ or τ whose corresponding vertex w is bifacial and incident to an inner face f of G_ν, we verify whether λ has labeled $+1$ the angle that w makes in f. If so, we replace the symbol σ or τ with σ_ℓ or τ_ℓ, respectively. By construction, Σ is the restriction of the signature of B to B'. ◀

The next result will be used to bound the number of interesting children of a P-node.

▶ **Lemma 7.** *Let ν be a P-node of T with poles u, v. Suppose that G_ν is plane, and let μ and μ' be two children of ν none of which is a Q-node, and whose corresponding edges of* $\text{skel}(\nu)^-$ *are consecutive in the permutation fixed by the planar embedding of G_ν. Also, suppose that for both μ and μ' it holds that the pair of signatures of its two half-boundaries is empty. Let*

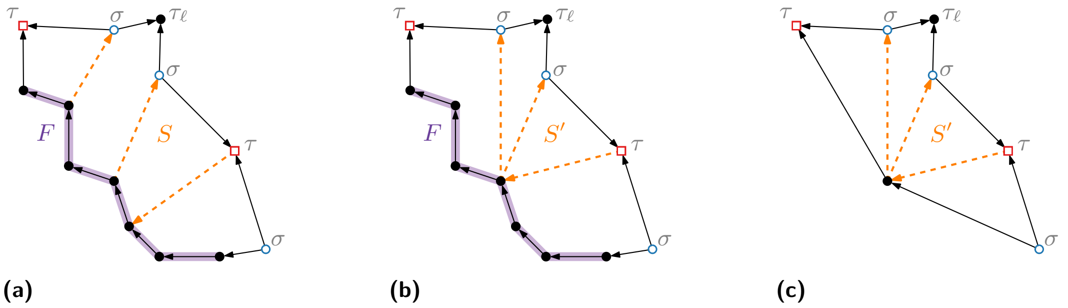


Figure 5 Illustration for Lemma 5.

G' be the digraph obtained from G by removing all vertices of $G_{\mu'}$ except the poles u, v. Then G is a YES-instance of st-PEC if and only if G' is a YES-instance.

Proof. Let F be the set of vertices of $G_{\mu'}$ distinct from u to v (i.e., those removed when going from G to G'). Also, suppose that the half-boundaries forming an inner face of G_ν are B_{vu} of μ and B_{uv} of μ'. Suppose first that G admits a solution, namely we can add a set E' of k saturating edges to G and turn it into an *st*-planar graph. If none of these edges is such that exactly one end-vertex belongs to F, then E' is a solution also for G'. Otherwise, let $S \in E'$ be the set of edges having exactly one end-vertex in F. Observe that the end-vertices in F all belong to the half-boundary B_{vu} of μ'. Hence, since neither μ nor μ' is a Q-node, analogously as in the proof of Lemma 5, we can replace S with a set S' of edges incident to an arbitrary vertex of the half-boundary B_{vu} of μ. This yields a new set E'' of saturating edges that corresponds to a solution for G'.

Suppose now that G' admits a solution E'. If E' does not contain any edge having exactly one end-vertex in B_{vu} of μ, then E' is a solution for G. Otherwise, let $S \in E'$ be the set of edges having exactly one end-vertex in B_{vu} of μ. again, since neither μ nor μ' is a Q-node, analogously as in the proof of Lemma 5, we can replace S with a set S' of edges incident to an arbitrary vertex of the half-boundary B_{vu} of μ'. This yields a new set E'' of saturating edges that corresponds to a solution for G. ◀

We are now ready to deal with P- and R-nodes.

▶ **Lemma 8.** *Let ν be a P-node of T. The set of partial solutions of ν can be computed in $2^{O(k^2)} \cdot n$ time.*

Proof. Let u and v be the poles of ν, and let $\mu_1, \mu_2, \ldots, \mu_h$ be the $h \geq 2$ children of ν. In order to compute the partial solutions for ν, similarly as for S-nodes, we check whether sets of internal assignments of G_{μ_1}, G_{μ_2}, …, G_{μ_h} can be combined together. For each child μ_i, let $\langle \Sigma_{1,i}, \Sigma_{2,i}, b_{1,i}, b_{2,i} \rangle$ be a candidate tuple. Let $C = \sum_{i=1}^{h} X(\mu_i, \Sigma_{1,i}, \Sigma_{2,i}, b_{1,i}, b_{2,i})$.

We first verify that $C \leq k$, and that at most one flag $b_{1,i}$ is true, as well as at most one flag $b_{2,i}$ is true. The first condition guarantees that we have not exceeded our budget k, while the second condition guarantees that the poles u, v shared by the children of ν do not receive the value $+1$ twice in the final internal assignment. If both conditions are satisfied, then we proceed as detailed below, otherwise we discard the set of candidate tuples.

Observe that h might be unbounded with respect to k, thus we cannot afford to enumerate all possible permutations of the edges of $\mathrm{skel}^-(\nu)$. To overcome this issue, we make the following crucial observations. First, we know that G contains at most $2k + 2$ switches, otherwise we can safely reject the instance. Consequently, at most $2k + 2$ children of ν may

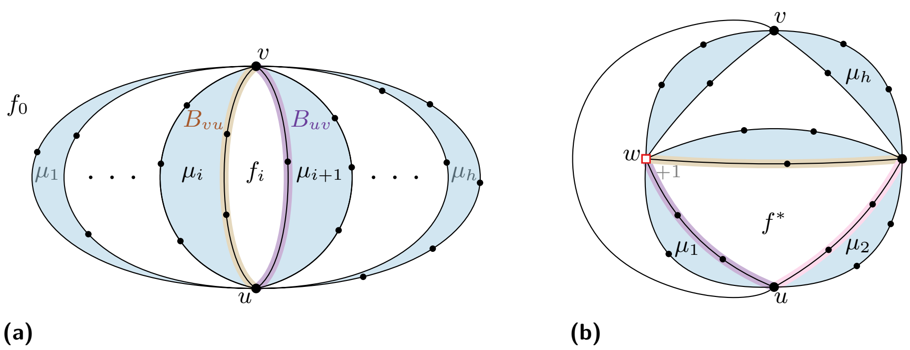


Figure 6 Illustration for the proof of (a) Lemma 8 and (b) Lemma 9.

contain switches different from u and v in their pertinent graphs. Second, consider now a permutation of the edges of $\text{skel}^-(\nu)$ and the corresponding planar embedding of G_ν. Up to a relabeling of the children, we shall assume that the half-boundary B_{vu} of μ_i and the half-boundary B_{uv} of μ_{i+1} form a face f_i of G_ν, for $i = 1, \dots, h-1$, and that the external face f_0 of G_ν consists of B_{uv} of μ_1 and B_{vu} of μ_h; see Figure 6a. Observe now that each of u and v can contribute at most one angle labeled $+1$ and at most two angles labeled 0; all other angles at u and v must be labeled -1. Hence, besides the at most six faces in which u or v contribute an angle labeled $+1$ or 0, all other faces are such that they either contain an angle labeled $+1$, or all their angles (except those formed by u and v) are labeled 0. Therefore, the number of faces whose half-boundaries have non-empty signatures is at most $t = 2(2k+2)+6 = 4k+10$ (a switch vertex may be bifacial and hence belong to two half-boundaries). Putting all together, if there exist more than t pairs that are not empty, then we can safely discard the set of candidate tuples.

Based on the previous observations, we will now assume to have at least $h - t$ empty pairs. Furthermore, if $h > 2t+2$, at least two children are such that Lemma 7 holds for them. Consequently, removing all empty pairs except $t+1$ preserves the existence of a solution (if any). Therefore, we shall further assume that we have $h \in O(t) \in O(k)$ pairs of signatures, and we can now enumerate all possible permutations of such pairs, and hence all possible putative planar embeddings described by $\text{skel}^-(\nu)$.

Consider now a fixed permutation. Following the same notation as before, assume that the half-boundary B_{vu} of μ_i and the half-boundary B_{uv} of μ_{i+1} form a face f_i of G_ν, for $i = 1, \dots, h$. We call such faces *active*. If all values $b_{1,i}$ are true and u is a switch, we guess whether u has an angle labeled $+1$ in some active face f_i or not. In the former case, we set flag b_1 to false and also guess which active face the angle belongs to, in the latter case we set b_1 to true. We do the same for v and flag b_2.

Next, consider a non-empty signature containing a symbol σ or τ. Let w be the vertex corresponding to that symbol. If w is not bifacial, the active face in which it forms the $+1$ angle is unique, otherwise we must guess in which of the two active faces sharing w the $+1$ angle is assigned to. After doing this procedure for all such symbols, we have exhaustively branched over the $2^{O(k)}$ angle assignments for the active faces. For each such angle assignment we can check, in $O(k)$ time, whether it is an upward assignment for each active face. If not, we discard the angle assignment, otherwise we now have an internal assignment λ of G_ν.

Next, for each active face f_i, we can apply Lemma 5 to compute the minimum number c_i of saturating edges needed to saturate all switches in f_i. Let $C + \sum_{i=1}^{h} c_i$ be the cost of the internal assignment λ. If it is larger than k, the angle assignment is discarded.

We are now ready to construct the signatures Σ_1 and Σ_2 of the half-boundaries B_{uv} and B_{vu} of ν. Since the half-boundary B_{uv} of ν coincides with B_{uv} of μ_1 (as fixed by the permutation at hand), we can invoke Lemma 6 by using λ and $\Sigma_{1,1}$ as arguments. Similarly, the signature Σ_2 is computed invoking Lemma 6 with arguments λ and $\Sigma_{2,h}$. Observe that both Σ_1 and Σ_2 are short, because $\Sigma_{1,1}$ and $\Sigma_{2,h}$ are short. Then we set $X(\Sigma_1, \Sigma_2, b_1, b_2) = \min\{X(\Sigma_1, \Sigma_2, b_1, b_2), C + \sum_{i=1}^{h} c_i\}$; taking the minimum is needed because different permutations, as well as different angle assignments of the same permutation, may yield the same pair of signatures and flags but different costs.

Putting all together, it suffices to first branch over sets of candidate tuples of size $h \in O(k)$, for each set we branch over $k^{O(k)}$ permutations, and for each permutation we further branch over the $2^{O(k)}$ possible angle assignments of the active faces. Computing the cost of an internal assignment takes $2^{O(k^2)} \cdot n$ time by using Lemma 5. ◀

▶ **Lemma 9.** *Let ν be an R-node of T. The set of partial solutions of ν can be computed in $2^{O(k^2)} \cdot n$ time.*

Proof. Let u and v be the poles of ν, and let $\mu_1, \mu_2, \ldots, \mu_h$ be the $h \geq 2$ children of ν. For each child μ_i, let $\langle \Sigma_{1,i}, \Sigma_{2,i}, b_{1,i}, b_{2,i} \rangle$ be a candidate tuple. Let $C = \sum_{i=1}^{h} X(\mu_i, \Sigma_{1,i}, \Sigma_{2,i}, b_{1,i}, b_{2,i})$.

We first verify that $C \leq k$, in order to avoid exceeding the budget. Next, we check the consistency of the flags. Recall that the vertices of skel(ν) are the poles of the children of ν. Namely, for each vertex w of skel(ν), we verify that at most one flag corresponding to it is false. If these conditions are met we proceed as detailed below, otherwise we discard the set of candidate tuples.

We now make important observations concerning the number of interesting children of ν. As in the proof of Lemma 8, we can observe that at most $2k + 2$ children of ν may contain switches different from u and v in their pertinent graphs. Now consider a child μ of ν that does not contain switches in its pertinent graph G_μ, and let u_μ and v_μ be its poles. If G admits a solution, one immediately verifies that G_μ is *st*-planar and its two switches are u_μ and v_μ. Consequently, in any solution, the two signatures $\Sigma_{u_\mu v_\mu}$ and $\Sigma_{v_\mu u_\mu}$ must be empty. Based on this property, it suffices to consider sets of pairs of signatures in which at most $2k + 2$ pairs are not empty.

Next, following the lines of the proof of Lemma 8, consider a non-empty signature containing a symbol σ or τ. Let w be the vertex corresponding to that symbol. If w is not bifacial, the face in which it forms the $+1$ angle is unique, otherwise we must guess in which of the two faces sharing w the $+1$ angle is assigned to. This is however not enough for R-nodes. Namely, observe that each face f^* of skel$(\nu)^-$ corresponds to a face f of G_ν whose boundary is formed by one half-boundary for each child of ν represented by an edge of f^* (which can be a real edge or a virtual edge); see Figure 6b. We call such faces *active* in the following. Moreover, the only angles that are not yet defined are those made by the vertices of skel(ν) that are switches and whose corresponding flags are all true. For these vertices we shall guess in which active face they make their $+1$ angle. Clearly, any such a vertex w belongs to multiple active faces (possibly including the external face). On the other hand, for an active face to be able to absorb a $+1$ angle, it must contain at least three angles labeled -1. Since we have at most $2k + 2$ non-empty pairs, there are at most $4k + 4$ active faces formed by non-empty signatures. For the other active faces, the only source of -1 angles are the vertices of skel(ν). Consequently, if w is incident to more than $4k + 5$ active faces in which the number of angles labeled -1 is larger than 2, we can safely discard the set of candidate tuples. Putting all together, for each vertex w we can branch over its $O(k)$ interesting active faces to decide in which of them it will make its $+1$ angle. This procedure leads to $2^{O(k)}$ angle assignments for the active faces. For each such angle assignment we can check, in $O(k)$ time, whether it is an upward assignment for each of the active faces. If not, we discard the angle assignment, otherwise we now have an internal assignment λ of G_ν.

Next, for each active face f_i, we can apply Lemma 5 to compute the minimum number c_i of saturating edges needed to saturate all switches in f_i. Let $C + \sum_{i=1}^{h} c_i$ be the cost of the internal assignment. If it is larger than k, the angle assignment is discarded.

We are now ready to construct the signatures Σ_1 and Σ_2 of the half-boundaries B_{uv} and B_{vu} of ν. Observe that the embedding of skel(ν) if fixed up to a flipping operation, which corresponds to inverting the two signatures. Therefore, we construct Σ_1 and Σ_2 as follows. Let Σ'_i, for $i = 1, \ldots, r$ be the $r \geq 1$ signatures of the half-boundaries of the children of ν that form the half-boundary B_{uv} of ν, in the order they are encountered from u to v. Also let w_i, $i = 1, \ldots, r-1$ be the vertices of skel(ν) that belong to B_{uv}. We initialize Σ_1 with the signature obtained by invoking Lemma 6 with arguments λ and Σ'_1. For vertex w_1, we distinguish whether it is a switch of G or not. In the former case, we concatenate the symbol σ (τ) if none of its angles in G_ν is labeled as $+1$, otherwise we concatenate σ_ℓ (τ_ℓ). In the latter case, consider the two edges incident to w_1 along B_{uv}. If one edge is incoming and the other is outgoing, then we do not append any symbol. If both edges are outgoing (incoming), we append σ_ℓ (τ_ℓ). We then repeat the procedure for the remaining signatures and vertices. The signature Σ_2 is computed analogously. Once both Σ_1 and Σ_2 have been computed, we verify that each of them is short (a necessary condition by Lemma 3), otherwise we reject the set of candidate tuples. Concerning the flags, b_1 (b_2) is true if and only if all flags corresponding to u (v) are true and none of its angles in the active faces is labeled $+1$ according to λ. Finally we set $X(\Sigma_1, \Sigma_2, b_1, b_2) = \min\{X(\Sigma_1, \Sigma_2, b_1, b_2), C + \sum_{i=1}^{h} c_i\}$, as well as $X(\Sigma_2, \Sigma_1, b_2, b_1) = \min\{X(\Sigma_2, \Sigma_1, b_2, b_1), C + \sum_{i=1}^{h} c_i\}$.

Putting all together, it suffices to first branch over sets of candidate tuples of size $h \in O(k)$, for each set we branch over the $2^{O(k)}$ possible angle assignments of the active faces. Computing the cost of an internal assignment takes $2^{O(k^2)} \cdot n$ time by using Lemma 5. ◀

It remains to deal with the root ρ of T. Recall that $G_\rho = G$, and that ρ is a Q-node.

▶ **Lemma 10.** *Let G be an n-vertex biconnected digraph, let e be an edge of G, and let $k \in \mathbb{N}$. There exists an algorithm that decides, in $O(2^{O(k^2)} \cdot n)$ time, whether G can be augmented to an st-planar graph with e on its external face by adding at most k edges.*

Proof. By using Lemmas 4, 8, and 9 we can traverse T bottom up until reaching the root ρ. Let u, v be the end-vertices of e, and therefore the poles of both ρ and of its child ξ. Consider each pair of signatures Σ'_1, Σ'_2 and of flags b_1, b_2 such that $C = X(\xi, \Sigma'_1, \Sigma'_2, b_1, b_2) \leq k$. Let f_0 and f_1 be the two faces of $G_\rho = G$ that share edge e, with f_0 being the external face. Without loss of generality, assume that the boundary of f_0 is formed by edge e and by the half-boundary B_{vu} of ξ, while the boundary of f_1 is formed by e and the half-boundary B_{uv} of ξ. Each symbol σ or τ in Σ_1 whose corresponding vertex is not bifacial must contribute with an angle labeled $+1$ in f_1. On the other hand, a symbol σ or τ in Σ'_1 whose corresponding vertex is bifacial must contribute with an angle labeled $+1$ in one of f_1 or f_0. For such vertices we guess in which of the two faces they contribute the $+1$ angle. We proceed analogously for Σ'_2. The same reasoning applies to u (v) if b_1 (b_2) is true. This leads to an $2^{O(k)}$ angle assignments for f_0 and f_1. For each angle assignment we can test, in $O(k)$ time, whether it is upward for the two faces f_0 and f_1.

An angle assignment of f_0 and f_1 that is upward for them, together with the internal assignment represented by $X(\xi, \Sigma'_1, \Sigma'_2, b_1, b_2)$, implies the existence of an upward angle assignment of G. Also, by Lemma 5 we can compute in $2^{O(k^2)} \cdot n$ time the cost of saturating

f_1. For face f_0 we need to saturate all switches except one source and one sink, hence its cost c_0 can be computed by a simple adjustment of the procedure of Lemma 5. If $C^* = C + c_0 + c_1 \leq k$, then we have constructed an upward angle assignment of G, and, in particular, all switches of G can be saturated with at most k saturating edges, except for a single source and a single sink on the external face. Then we can conclude that G is a YES instance and the algorithm reports a positive answer.

After considering all pairs Σ'_1, Σ'_2 and all pairs b_1, b_2, as well as all angle assignments for the corresponding faces f_0 and f_1, if no positive answer was returned, then the algorithm halts and rejects the instance. ◀

The proof of Theorem 1 follows by applying Lemma 10 for each of the $O(n)$ edges of G.

5 Discussion and Open Problems

We showed that *st*-PEC can be solved in $2^{O(k^2)} \cdot n^2$ time for biconnected digraphs. It is worth remarking that, while in principle the *st*-PEC problem needs not to be restricted to biconnected digraphs (for which it is already NP-hard), considering simply connected graphs would make the proof of our result more technical but not more interesting. In fact, one can simply decompose the graph into its biconnected components through a block-cutvertex tree and work with similar boundary conditions as those we already considered. More interestingly, we ask whether *st*-PEC belongs to the FPL (fixed parameter linear) class. On a similar note, improving the exponential function (or proving that it is asymptotically optimal under standard assumptions) would also be interesting. Lastly, it remains open whether *st*-PEC admits a kernel of polynomial size.

References

1. Paola Bertolazzi, Giuseppe Di Battista, and Walter Didimo. Quasi-upward planarity. *Algorithmica*, 32(3):474–506, 2002.
2. Paola Bertolazzi, Giuseppe Di Battista, Giuseppe Liotta, and Carlo Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(6):476–497, 1994.
3. Carla Binucci, Emilio Di Giacomo, Giuseppe Liotta, and Alessandra Tappini. Quasi-upward planar drawings with minimum curve complexity. In *GD*, volume 12868 of *Lecture Notes in Computer Science*, pages 195–209. Springer, 2021.
4. Pablo Burzyn, Flavia Bonomo, and Guillermo Durán. NP-completeness results for edge modification problems. *Discret. Appl. Math.*, 154(13):1824–1844, 2006. URL: `https://doi.org/10.1016/j.dam.2006.03.031`, `doi:10.1016/j.dam.2006.03.031`.
5. Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Robert Ganian, Chrysanthi N. Raftopoulou, and Kirill Simonov. Parameterized algorithms for upward planarity. In *SoCG*, volume 224 of *LIPIcs*, pages 26:1–26:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
6. Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Robert Ganian, Chrysanthi N. Raftopoulou, and Kirill Simonov. Testing upward planarity of partial 2-trees. In *GD*, volume 13764 of *Lecture Notes in Computer Science*, pages 175–187. Springer, 2022.
7. Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr A. Golovach. A survey of parameterized algorithms and the complexity of edge modification. *Comput. Sci. Rev.*, 48:100556, 2023. URL: `https://doi.org/10.1016/j.cosrev.2023.100556`, `doi:10.1016/j.cosrev.2023.100556`.
8. Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.

9 Giuseppe Di Battista, Giuseppe Liotta, and Francesco Vargiu. Spirality and optimal orthogonal drawings. *SIAM J. Comput.*, 27(6):1764–1811, 1998.

10 Giuseppe Di Battista and Roberto Tamassia. Algorithms for plane representations of acyclic digraphs. *Theor. Comput. Sci.*, 61:175–198, 1988.

11 Walter Didimo, Francesco Giordano, and Giuseppe Liotta. Upward spirality and upward planarity testing. *SIAM J. Discret. Math.*, 23(4):1842–1899, 2009.

12 Walter Didimo, Michael Kaufmann, Giuseppe Liotta, and Giacomo Ortali. Rectilinear planarity of partial 2-trees. In *GD*, volume 13764 of *Lecture Notes in Computer Science*, pages 157–172. Springer, 2022.

13 Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001.

14 Carsten Gutwenger and Petra Mutzel. A linear time implementation of SPQR-trees. In *GD 2000*, volume 1984 of *LNCS*, pages 77–90. Springer, 2001. URL: https://doi.org/10.1007/3-540-44541-2.

15 John E. Hopcroft and Robert Endre Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973. URL: https://doi.org/10.1137/0202012.

16 Assaf Natanzon, Ron Shamir, and Roded Sharan. Complexity classification of some edge modification problems. *Discret. Appl. Math.*, 113(1):109–128, 2001. URL: https://doi.org/10.1016/S0166-218X(00)00391-7, doi:10.1016/S0166-218X(00)00391-7.

17 William T. Trotter and John I. Moore Jr. The dimension of planar posets. *J. Comb. Theory, Ser. B*, 22(1):54–67, 1977.