Graph Planarity Testing with Hierarchical Embedding Constraints*

Giuseppe Liotta[1], Ignaz Rutter[2], and Alessandra Tappini[1]

1 **Dipartimento di Ingegneria, Università degli Studi di Perugia, Italy**
giuseppe.liotta@unipg.it, alessandra.tappini@studenti.unipg.it

2 **Department of Computer Science and Mathematics, University of Passau, Germany**
rutter@fim.uni-passau.de

Abstract

Hierarchical embedding constraints define a set of allowed cyclic orders for the edges incident to the vertices of a graph. These constraints are expressed in terms of FPQ-trees. FPQ-trees are a variant of PQ-trees that includes F-nodes in addition to P-nodes and to Q-nodes: An F-node represents a permutation that is fixed, i.e., it cannot be reversed. Let G be a graph such that every vertex of G is equipped with a set of FPQ-trees encoding hierarchical embedding constraints for its incident edges. We study the problem of testing whether G admits a planar embedding such that, for each vertex v of G, the cyclic order of the edges incident to v is described by at least one of the FPQ-trees associated with v. We prove that the problem is fixed-parameter tractable for biconnected graphs, where the parameters are the treewidth of G and the number of FPQ-trees associated with every vertex. We also show that the problem is NP-complete if parameterized by the number of FPQ-trees only, and W[1]-hard if parameterized by the treewidth only.

1 Introduction

The study of graph planarity testing and of its variants is at the heart of graph algorithms and of their applications. This paper is inspired by a work of Gutwenger et al. [7], who study the graph planarity testing problem subject to hierarchical embedding constraints. Hierarchical embedding constraints specify for each vertex v of G which cyclic orders of the edges incident to v are admissible in a constrained planar embedding of G. For example, Fig. 1 shows the edges incident to a vertex v and a set of hierarchical embedding constraints on these edges. Edges are partitioned into four sets, denoted as E_1, E_2, E_3, and E_4; the constraints allow only two distinct clockwise cyclic orders for these edge-sets, namely either $E_1E_2E_3E_4$ (Fig. 1a) or $E_1E_3E_2E_4$ (Fig. 1b). Within each set, the constraints of Fig. 1 allow the edges of E_1, E_2, and E_3 to be arbitrarily permuted, while the edges of E_4 are partitioned into three subsets E_4', E_4'', and E_4''' such that E_4'' must appear between E_4' and E_4''' in the clockwise order around v. The edges of E_4' can be arbitrarily permuted, while the edges of E_4'' and the edges of E_4''' have only two possible orders that are the reverse of one another.

Hierarchical embedding constraints can be encoded by using FPQ-trees, a variant of PQ-trees that includes F-nodes in addition to P-nodes and to Q-nodes. An F-node encodes a permutation that cannot be reversed. For example, the hierarchical embedding constraints of Fig. 1 can be represented by two FPQ-trees denoted as T and T' in Fig. 1a and 1b.

Gutwenger et al. [7] study the planarity testing problem with hierarchical embedding

* Work partially supported by: ***(i)*** MIUR, under grant 20174LF3T8 "AHeAD: efficient Algorithms for HArnessing networked Data"; ***(ii)*** Dipartimento di Ingegneria - Università degli Studi di Perugia, under grant RICBA19FM: "Modelli, algoritmi e sistemi per la visualizzazione di grafi e reti"; ***(iii)*** German Science Foundation (DFG), under grant Ru 1903/3-1.

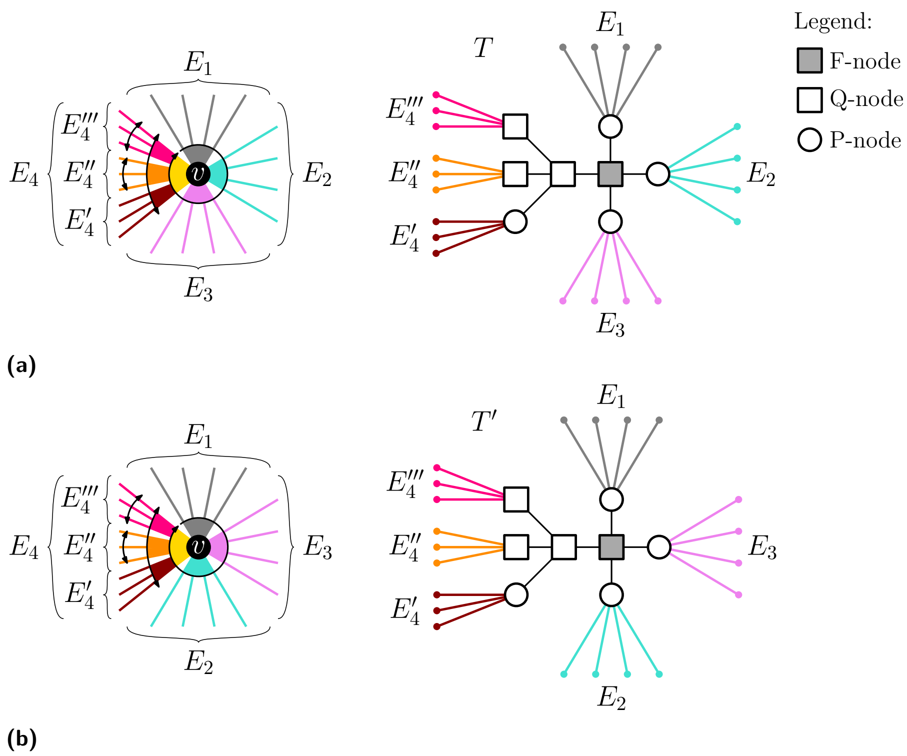

Figure 1 Two examples of a vertex v with hierarchical embedding constraints and the corresponding FPQ-trees. F-nodes are shaded boxes, Q-nodes are white boxes, and P-nodes are circles.

constraints by allowing *at most one* FPQ-tree per vertex. We generalize their study and allow *more than one* FPQ-tree associated with each vertex. Our main results are the following.

- We show that FPQ-Choosable Planarity Testing is NP-complete even if the number of FPQ-trees associated with each vertex is bounded by a constant greater than 1, and it remains NP-complete even if the FPQ-trees only contain P-nodes. This contrasts with the result of Gutwenger et al. [7] who prove that FPQ-Choosable Planarity Testing can be solved in linear time when each vertex is equipped with at most one FPQ-tree.
- We prove that FPQ-Choosable Planarity Testing is W[1]-hard if parameterized by treewidth, and that it remains W[1]-hard even when the FPQ-trees only contain P-nodes.
- The above results imply that FPQ-Choosable Planarity Testing is not fixed-parameter tractable if parameterized by treewidth only or by the number of FPQ-trees per vertex only. For a contrast, we show that FPQ-Choosable Planarity Testing becomes fixed-parameter tractable for biconnected graphs when parameterized by both the treewidth and the number of FPQ-trees associated with every vertex.

Proofs and details omitted from this extended abstract can be found in the full version [8].

Preliminaries. We assume familiarity with graph theory and algorithms, and with the concepts of PQ-tree, SPQR-decomposition tree, branchwidth, treewidth and sphere-cut decomposition of a graph [3, 4, 5, 6, 9]. We only briefly recall some of the basic concepts that will be used extensively in the rest of the paper (see also [1]).

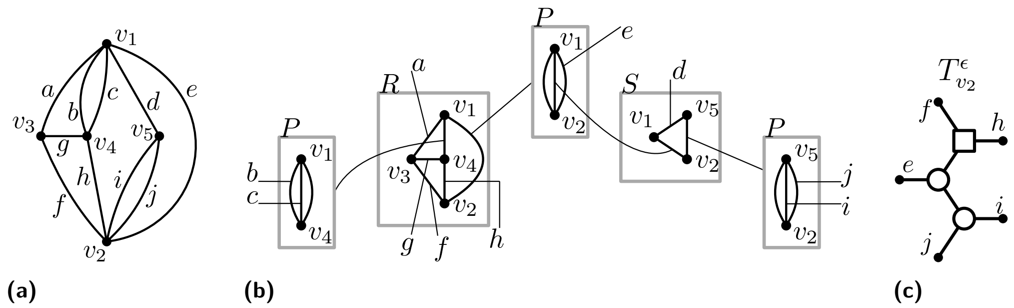

Figure 2 (a) A biconnected planar graph G. (b) An SPQR-decomposition tree of G. (c) The embedding tree of v_2.

Given a graph G together with a fixed combinatorial embedding, we can associate with each vertex v a PQ-tree T_v whose leaves represent the edges incident to v. The tree T_v encodes a set of permutations for its leaves: Each of these permutations is in a bijection with a cyclic order of the edges incident to v. If there is a permutation π_v of the leaves of T_v that is in a bijection with a cyclic order σ_v of the edges incident to v, we say that T_v *represents* σ_v, or equivalently that σ_v *is represented by* T_v. An *FPQ-tree* is a PQ-tree where, for some of the Q-nodes, the reversal of the permutation described by their children is not allowed. To distinguish these Q-nodes from the regular ones, we call them *F-nodes*.

The planar combinatorial embeddings that are given by the SPQR-decomposition tree of a biconnected graph G give constraints on the cyclic order of edges around each vertex of G. These constraints can be encoded by associating a PQ-tree with each vertex v of G, called the *embedding tree* of v and denoted by T_v^ϵ (see, e.g., [2]). For example, Fig. 2c shows the embedding tree $T_{v_2}^\epsilon$ of the vertex v_2 in Fig. 2a. Note that edges f and h (i and j, resp.) belong to an R-node (a P-node, resp.) in the SPQR-decomposition tree of G (Fig. 2b), hence the corresponding leaves are connected to a Q-node (a P-node, resp.) in $T_{v_2}^\epsilon$.

2 The FPQ-choosable Planarity Testing Problem

Let $G = (V, E)$ be a (multi-)graph, let $v \in V$, and let T_v be an FPQ-tree whose leaf set is $E(v)$. We define *consistent*(T_v) as the set of cyclic orders of the edges incident to v in a planar embedding $\mathcal{E}$ of G that are represented by the FPQ-tree T_v. An *FPQ-choosable graph* is a pair (G, D) where $G = (V, E)$ is a (multi-)graph, and D is a mapping that associates each vertex $v \in V$ with a set $D(v)$ of FPQ-trees whose leaf set is $E(v)$. Given a planar embedding $\mathcal{E}$ of G, we denote by $\mathcal{E}(v)$ the cyclic order of edges incident to v in $\mathcal{E}$. An *assignment* A is a function that assigns to each vertex $v \in V$ an FPQ-tree in $D(v)$. We say that A is *compatible with* G if there exists a planar embedding $\mathcal{E}$ of G such that $\mathcal{E}(v) \in$ *consistent*$(A(v))$ for all $v \in V$. In this case, we also say that $\mathcal{E}$ is *consistent with* A. An FPQ-choosable graph (G, D) is *FPQ-choosable planar* if there exists an assignment that is compatible with G. Refer to Fig. 3 for an example.

The FPQ-Choosable Planarity Testing problem receives as input an FPQ-choosable graph (G, D) and it asks whether (G, D) is FPQ-choosable planar. In the rest of the paper we assume that G is a biconnected (multi-)graph. Clearly G must be planar or else the problem becomes trivial. Also, any assignment that is compatible with G must define a planar embedding of G among those described by an SPQR-decomposition tree of G. Therefore, a preliminary step for an algorithm that tests whether (G, D) is FPQ-choosable

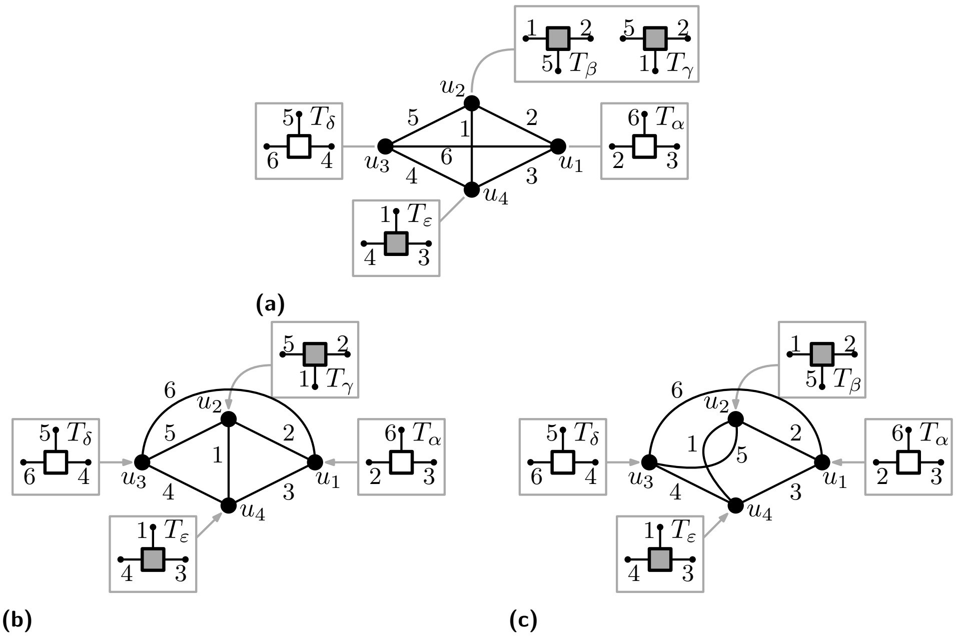

Figure 3 (a) An FPQ-choosable planar graph (G, D). (b) A planar embedding of G that is consistent with assignment $\{A(u_1) = T_\alpha, A(u_2) = T_\gamma, A(u_3) = T_\delta, A(u_4) = T_\varepsilon\}$; the assignment is compatible with G. (c) A non-planar embedding of G that is consistent with assignment $\{A(u_1) = T_\alpha, A(u_2) = T_\beta, A(u_3) = T_\delta, A(u_4) = T_\varepsilon\}$; there is no planar embedding that is consistent with A.

planar is to intersect each FPQ-tree $T_v \in D(v)$ with the embedding tree T_v^ϵ of v, so that the cyclic order of the edges incident to v satisfies both the constraints given by T_v and the ones given by T_v^ϵ. (See, e.g., [2] for details about the operation of intersection between two PQ-trees, whose extension to the case of FPQ-trees is straightforward). We assume that the FPQ-trees of D have been intersected with the corresponding embedding trees and we still denote by $D(v)$ the set of FPQ-trees associated with v and resulting from the intersection.

3 Complexity of FPQ-choosable Planarity Testing

As we are going to show, FPQ-Choosable Planarity Testing is fixed-parameter tractable when parameterized by treewidth and number of FPQ-trees per vertex. One may wonder whether the problem remains FPT if parameterized by the treewidth only or by the number of FPQ-trees per vertex only. The following theorems answer this question in the negative.

▶ **Theorem 3.1.** FPQ-Choosable Planarity Testing *with a bounded number of FPQ-trees per vertex is NP-complete. It is NP-complete even if the FPQ-trees have only P-nodes.*

▶ **Theorem 3.2.** FPQ-Choosable Planarity Testing *parameterized by treewidth is W[1]-hard. It is W[1]-hard even if the FPQ-trees have only P-nodes.*

4 Fixed Parameter Tractability of FPQ-choosable Planarity Testing

In this section, we introduce some concepts that are fundamental to the description of the algorithm and we present a polynomial-time testing algorithm for graphs having bounded branchwidth and such that the number of FPQ-trees associated with each vertex is bounded by a constant. Note that, for a graph G with treewidth t and branchwidth $b > 1$, it holds that $b - 1 \leq t \leq \left\lfloor \frac{3}{2}b \right\rfloor - 1$ [9].

Let T be an FPQ-tree, let leaves(T) denote the set of its leaves, and let L be a proper subset of leaves(T). We denote by σ a cyclic order of the leaves of an FPQ-tree, and we say that $\sigma \in$ *consistent*(T) if the FPQ-tree T represents σ. We say that L is a *consecutive set* if the leaves in L are consecutive in every cyclic order represented by T. Let e be an edge of T, and let T' and T'' be the two subtrees obtained by removing e from T. If either leaves(T') or leaves(T'') are a subset of a consecutive set L, then we say that e is a *split edge for* L. The subtree that contains the leaves in L is the *split subtree* of e for L. A split edge e is *maximal* for L if there exists no split edge e' such that the split subtree of e' contains e.

▶ **Lemma 4.1.** *Let T be an FPQ-tree, let L be a consecutive proper subset of* leaves(T)*, and let S be the set of maximal split edges for L. Then either $|S| = 1$, or $|S| > 1$ and there exists a Q-node (or an F-node) χ of T such that χ has degree at least $|S| + 2$ and the elements of S appear consecutive around χ.*

If $|S| = 1$, the split edge in S is called the *boundary of* L. If $|S| > 1$, the Q-node (or F-node) χ defined in the statement of Lemma 4.1 is the *boundary of* L. Since F-nodes are a more constrained version of Q-nodes, when we refer to boundary Q-nodes we also take into account the case of F-nodes. Fig. 4a shows an FPQ-choosable graph (G, D) and two FPQ-trees $T_u \in D(u)$ and $T_v \in D(v)$. The three red edges b, c, and d of G define a consecutive set L_u in T_u; the edges e and f define a consecutive set L_v in T_v. The boundary of L_u in T_u is a Q-node, while the boundary of L_v in T_v is an edge.

We denote by $\mathcal{B}(L)$ the boundary of a set of leaves L. If $\mathcal{B}(L)$ is a Q-node, we associate $\mathcal{B}(L)$ with a default orientation that arbitrarily defines one of the two possible permutations of its children. This default orientation is called the *clockwise orientation* of $\mathcal{B}(L)$, while the other possible permutation of the children of $\mathcal{B}(L)$ is the *counter-clockwise orientation*.

Let $L' = L \cup \{\ell\}$, where ℓ is a new element. Let $\sigma \in$ *consistent*(T), and let $\sigma|_{L'}$ be a cyclic order obtained from σ by replacing the elements of the consecutive set leaves(T)$\setminus L$ by the single element ℓ. We say that a cyclic order σ' of L' is *extensible* if there exists a cyclic order $\sigma \in$ *consistent*(T) with $\sigma|_{L'} = \sigma'$ (and σ is an *extension of* σ'). An extensible order σ is *clockwise* if the orientation of χ is clockwise; σ is *counter-clockwise* otherwise. If the boundary of L is an edge, we consider any extensible order as both clockwise and counter-clockwise.

Let (G, D) be an FPQ-choosable graph, let $\mathcal{T}$ be an SPQR-decomposition tree of G and let v be a pole of a node μ of $\mathcal{T}$, let $T_v \in D(v)$ be an FPQ-tree associated with v, let E_{ext} be the set of edges that are incident to v and not contained in the pertinent graph G_μ, and let $E^\star_\mu(v) = E(v) \setminus E_{\text{ext}}$. Note that there is a bijection between the edges $E(v)$ of G and the leaves of T_v, hence we shall refer to the set of leaves of T_v as $E(v)$. Also note that $E^\star_\mu(v)$ is represented by a consecutive set of leaves in T_v, because in every planar embedding of G the edges in $E^\star_\mu(v)$ must appear consecutively in the cyclic order of the edges incident to v.

The *pertinent FPQ-tree* of T_v, denoted as $\text{Pert}_\mu(T_v)$, is the FPQ-tree obtained from T_v by replacing the consecutive set E_{ext} with a single leaf ℓ. Informally, the pertinent FPQ-tree of v describes the hierarchical embedding constraints for v within G_μ. For example, in Fig. 4b a pertinent graph G_μ with poles u and v is highlighted by a shaded region; the pertinent FPQ-tree $\text{Pert}_\mu(T_u)$ of T_u and the pertinent FPQ-tree $\text{Pert}_\mu(T_v)$ of T_v are obtained by the FPQ-trees T_u and T_v of Fig. 4a.

Let $\nu_1, \ldots, \nu_k$ be the children of μ in $\mathcal{T}$. Observe that the edges $E^\star_{\nu_i}(v)$ of each G_{ν_i} $(1 \leq i \leq k)$ form a consecutive set of leaves of $A_\mu(v) = \text{Pert}_\mu(T_v)$. The *skeletal FPQ-tree* of $\text{Pert}_\mu(T_v)$, denoted by $\text{Skel}_\mu(T_v)$, is the tree obtained from $\text{Pert}_\mu(T_v)$ by replacing each of the

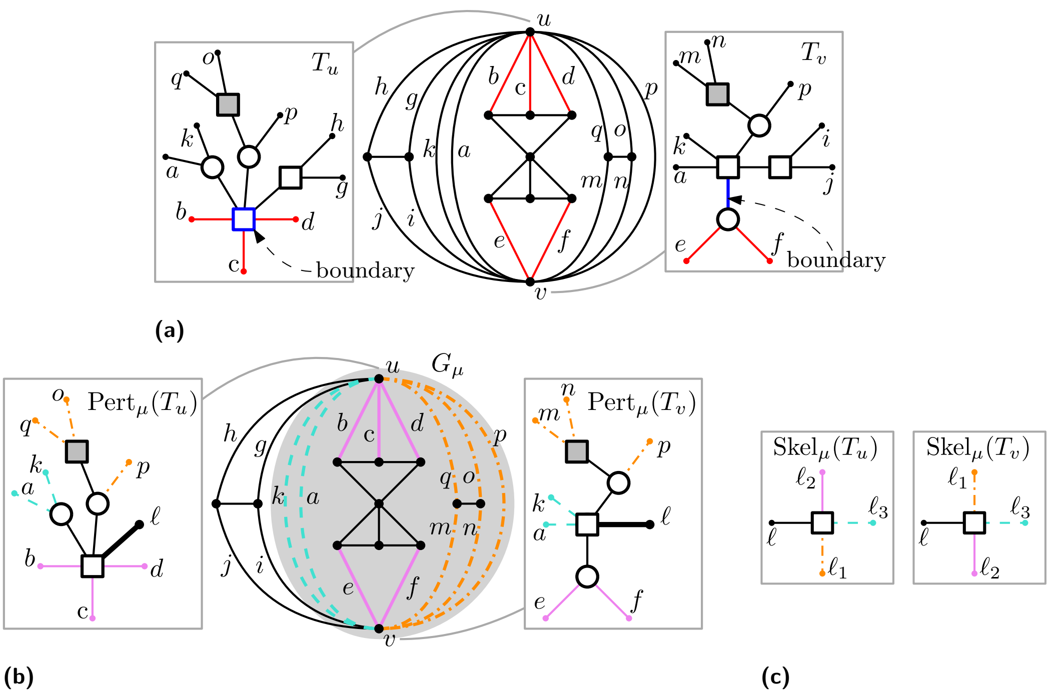

Figure 4 (a) A boundary Q-node in T_u and a boundary edge in T_v. (b) Pertinent FPQ-trees $\mathrm{Pert}_\mu(T_u)$ and $\mathrm{Pert}_\mu(T_v)$. (c) Skeletal FPQ-trees $\mathrm{Skel}_\mu(T_u)$ of $\mathrm{Pert}_\mu(T_u)$ and $\mathrm{Skel}_\mu(T_v)$ of $\mathrm{Pert}_\mu(T_v)$.

consecutive sets $E^\star_{\nu_i}(v)$ $(1 \le i \le k)$ by a single leaf ℓ_i (see Fig. 4c). Note that each Q-node of $\mathrm{Skel}_\mu(T_u)$ corresponds to a Q-node of $\mathrm{Pert}_\mu(T_u)$, and thus to a Q-node of T_u; also, distinct Q-nodes of $\mathrm{Skel}_\mu(T_u)$ correspond to distinct Q-nodes of $\mathrm{Pert}_\mu(T_u)$, and thus to distinct Q-nodes of T_u. For each Q-node χ of T_u that is a boundary of μ or of one of its children, there is a corresponding Q-node in $\mathrm{Skel}_\mu(T_u)$ that inherits its default orientation from T_u.

Let (G, D) be an FPQ-choosable graph, let $\mathcal{T}$ be an SPQR-decomposition tree of G, let μ be a node of $\mathcal{T}$, and let u and v be the poles of μ. We denote by (G_μ, D_μ) the FPQ-choosable graph consisting of the pertinent graph G_μ and the set D_μ that is defined as follows: $D_\mu(z) = D(z)$ for each vertex z of G_μ that is not a pole, and $D_\mu(v) = \{\mathrm{Pert}_\mu(T_v) \mid T_v \in D(v)\}$ if v is a pole of μ. A tuple $\langle T_u, T_v, o_u, o_v\rangle \in D(u) \times D(v) \times \{0,1\} \times \{0,1\}$ is *admissible for* G_μ if there exist an assignment A_μ of (G_μ, D_μ) and a planar embedding $\mathcal{E}_\mu$ of G_μ consistent with A_μ such that $A_\mu(u) = \mathrm{Pert}_\mu(T_u)$, $A_\mu(v) = \mathrm{Pert}_\mu(T_v)$, $\mathcal{B}(E^\star_\mu(u))$ is clockwise (counter-clockwise) in T_u if $o_u = 0$ $(o_u = 1)$, and $\mathcal{B}(E^\star_\mu(v))$ is clockwise (counter-clockwise) in T_v if $o_v = 0$ $(o_v = 1)$. A tuple is *admissible for* μ if it is admissible for G_μ. $\Psi(\mu)$ is the set of admissible tuples for G_μ.

FPT Algorithm: In order to test if (G, D) is FPQ-choosable planar, we root the SPQR-decomposition tree $\mathcal{T}$ at an arbitrary Q-node and we visit $\mathcal{T}$ from the leaves to the root. At each step of the visit, we equip the current node μ with the set $\Psi(\mu)$. If we encounter a node μ such that $\Psi(\mu) = \emptyset$, we return that (G, D) is not FPQ-choosable planar; otherwise the planarity test returns an affirmative answer. If the currently visited node μ is a leaf of $\mathcal{T}$, we set $\Psi(\mu) = D(u) \times D(v) \times \{0,1\} \times \{0,1\}$, because its pertinent graph is a single edge. If μ is an internal node, $\Psi(\mu)$ is computed from the sets of admissible tuples of the children of μ and depending on whether μ is an S-, P-, or R-node. In the case of R-nodes, we compute the set of admissible tuples by executing the sphere-cut decomposition of the skeleton of μ and by exploiting the fact that it has branchwidth at most b, where b is the branchwidth of G.

▶ **Theorem 4.2.** *Let* (G, D) *be a biconnected FPQ-choosable (multi-)graph such that* $G = (V, E)$ *and* $|V| = n$*. Let* $D(v)$ *be the set of FPQ-trees associated with vertex* $v \in V$*. There exists an* $O(D_{\max}^{\frac{3}{2}b} \cdot n^2 + n^3)$*-time algorithm to test whether* (G, D) *is FPQ-choosable planar, where* b *is the branchwidth of* G *and* $D_{\max} = \max_{v \in V} |D(v)|$*.*

As future work, it would be nice to extend Theorem 4.2 to simply connected graphs. Indeed, our proof is based on the SPQR-decomposition that assumes the biconnectivity of the input graph.

References

1. Subramanian Arumugam, Andreas Brandstädt, Takao Nishizeki, and Krishnaiyan Thulasiraman. *Handbook of graph theory, combinatorial optimization, and algorithms*. Chapman and Hall/CRC, 2016.
2. Thomas Bläsius and Ignaz Rutter. Simultaneous PQ-ordering with applications to constrained embedding problems. *ACM Trans. Algorithms*, 12(2):16:1–16:46, 2016.
3. Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.*, 13(3):335–379, 1976.
4. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.
5. Frederic Dorn, Eelko Penninkx, Hans L. Bodlaender, and Fedor V. Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut decompositions. *Algorithmica*, 58(3):790–810, 2010.
6. Qian-Ping Gu and Hisao Tamaki. Optimal branch-decomposition of planar graphs in $O(n^3)$ time. *ACM Trans. Algorithms*, 4(3):30:1–30:13, 2008.
7. Carsten Gutwenger, Karsten Klein, and Petra Mutzel. Planarity testing and optimal edge insertion with embedding constraints. *J. Graph Algorithms Appl.*, 12(1):73–95, 2008.
8. Giuseppe Liotta, Ignaz Rutter, and Alessandra Tappini. Graph planarity testing with hierarchical embedding constraints. *CoRR*, abs/1904.12596, 2019. `arXiv:1904.12596`.
9. Neil Robertson and Paul D. Seymour. Graph minors. X. obstructions to tree-decomposition. *J. Comb. Theory, Ser. B*, 52(2):153–190, 1991.