A Polynomial Delay Algorithm Generating All Potential Maximal Cliques in Triconnected Planar Graphs

Alexander Grigoriev
Department of Data Analytics and Digitalisation, Maastricht University, The Netherlands

Yasuaki Kobayashi
Hokkaido University, Sapporo, Japan

Hisao Tamaki
Meiji University, Kawasaki, Japan

Tom C. van der Zanden
Department of Data Analytics and Digitalisation, Maastricht University, The Netherlands

Abstract

We develop a new characterization of potential maximal cliques of a triconnected planar graph and, using this characterization, give a polynomial delay algorithm generating all potential maximal cliques of a given triconnected planar graph. Combined with the dynamic programming algorithm due to Bouchitté and Todinca, this algorithm leads to a treewidth algorithm for general planar graphs that runs in time linear in the number of potential maximal cliques and polynomial in the number of vertices.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Theory of computation → Graph algorithms analysis

Keywords and phrases potential maximal cliques, treewidth, planar graphs, triconnected planar graphs, polynomial delay generation

Digital Object Identifier 10.4230/LIPIcs.IPEC.2025.21

Funding *Yasuaki Kobayashi*: JSPS KAKENHI Grant Numbers JP23K28034, JP24H00686 and JP24H00697
Hisao Tamaki: JSPS KAKENHI Grant Number JP24H00697

1 Introduction

Let G be a graph. A vertex set X of G is a *potential maximal clique* (PMC) of G if there is a minimal triangulation H of G such that X is a maximal clique of H. Let $\Pi(G)$ denote the set of all PMCs of G. Computing $\Pi(G)$ is the first step in the treewidth algorithms of Bouchitté and Todinca [7]. The second step is a dynamic programming algorithm that works on $\Pi(G)$ and computes the treewidth $tw(G)$ of G, together with a tree-decomposition of G achieving the width, in time $|\Pi(G)|n^{O(1)}$. The same approach works for various problems that can be formulated as asking for an optimal triangulation with some criterion, including the minimum fill-in problem [7]. See [11] and [12] for more applications of PMCs in this direction.

Naturally, the complexity of computing $\Pi(G)$ is of great interest. In a separate paper [8], Bouchitté and Todinca showed that $\Pi(G)$ can be computed in time polynomial in the number of minimal separators (see Section 2 for the definition) of G. Fomin and Villanger [13] showed that $|\Pi(G)|$ is $O(1.7549^n)$ and gave an algorithm to compute $\Pi(G)$ in time $O(1.7549^n)$. (These bounds were later improved to $O(1.7346^n)$ in [12].) Although this bound on the running time matches the combinatorial bound, an algorithm with running time $|\Pi(G)|n^{O(1)}$

20th International Symposium on Parameterized and Exact Computation (IPEC 2025).
Editors: Akanksha Agrawal and Erik Jan van Leeuwen; Article No. 21; pp. 21:1–21:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is much more desirable in practice. In fact, the success of recent practical algorithms for treewidth based on the Bouchitté-Todinca algorithm, [20] for example, relies on the fact that $|\Pi(G)|$ is vastly smaller than this theoretical bound on many instances of interest in practice.

In this paper, we address this question if $\Pi(G)$ can be computed in time $|\Pi(G)|n^{O(1)}$, which is one of the open questions in parameterized and exact computation listed in [4]. We note that this question has been open for any natural graph class except those for which $\Pi(G)$ is known to be computable in polynomial time. We answer this question in the affirmative for the class of triconnected planar graphs. As a straightforward consequence, we obtain a new running time upper bound of $|\Pi(G)|n^{O(1)}$ on the treewidth computation, not only for triconnected planar graphs but also for general planar graphs, since the treewidth of a planar graph can be computed by working separately on its triconnected components.

This result is also interesting with regard to a broader question: how planarity helps in treewidth computation? In the case of branchwidth, a graph parameter similar to treewidth, there is a clear answer to the similar question. Branchwidth is NP-hard to compute for general graphs but polynomial time computable for planar graphs by the celebrated Ratcatcher algorithm due to Seymour and Thomas [19]. It has been a long open question whether an analogy holds for treewidth: we do not know whether computing treewidth is NP-hard or polynomial time solvable on planar graphs. In addition to the 1.5-approximation algorithm based on the Ratcatcher algorithm for branchwidth, several approximation algorithms for treewidth are known that exploit planarity ([17] for example, also see citations therein), but no non-trivial exact algorithm for planar treewidth is known. Several fixed-parameter tractable exact treewidth algorithms for general graphs are known, including a $2^{O(k^3)}n$ time algorithm [3] and a $2^{O(k^2)}n^{O(1)}$ time algorithm [18]. No improvement, however, on these results exploiting planarity is known.

Our algorithm is the first one that explicitly exploits planarity for computing exact treewidth in a non-trivial manner.

Our algorithm not only computes $\Pi(G)$ for triconnected planar graph G in time $|\Pi(G)|n^{O(1)}$ but also runs with polynomial delay: it spends $n^{O(1)}$ time for the generation of each element in the set. See Section 4 for a more technical definition of polynomial delay generation. This feature of our algorithm is appealing in the field of combinatorial enumeration where designing an algorithm with polynomial delay is one of the main research goals. We note that most of the devices we use in our algorithm are still needed even if we downgrade our requirement of polynomial delay to the total running time bound of $|\Pi(G)|n^{O(1)}$.

Our generation algorithm is based on a new and simple characterization of PMCs of triconnected planar graphs. Fomin, Todinca, and Villanger [10] used a characterization of PMCs of general planar graphs in terms of what they call *θ-structures*. A θ-structure is, roughly speaking, a collection of three curves sharing their ends in the sphere of embedding such that each pair forms a *noose* of the embedded graph, where a noose is a simple closed curve that intersects the embedded graph only at vertices. They used this characterization to design an algorithm for the maximum induced planar subgraph problem. Unfortunately for our purposes, this characterization does not seem to lead to an efficient algorithm for computing $\Pi(G)$ of planar graphs.

The simplicity of our characterization comes from the use of what we call a *latching graph* of a triconnected plane graph instead of the traditional tools for reasoning about separators in plane graphs: nooses or radial graphs, which are bipartite plane graphs representing the incidences of the vertices and faces (for a precise definition, see [16] for example). The latching graph L_G of a biconnected plane graph G is a multigraph obtained from G by adding, in each face of G, every chord of the bounding cycle of the face. We observe that L_G is simple if G is triconnected (Proposition 6). Separators of G correspond to cycles of L_G in a similar way as they correspond to cycles of the radial graph of G. Unlike a radial graph, however, that is a bipartite graph on the vertices and the faces of the embedded graph G, a latching graph is a graph on $V(G)$ and simpler to reason with for some purposes. More specifically, we define a class of biconnected plane graphs we call *steerings* (Definition 13 in Section 3) and show that a vertex set X of a triconnected plane graph G is a PMC if and only if the subgraph of L_G induced by X is a steering. This characterization would be less simple if it was formulated in terms of radial graphs or nooses. It is not clear if the latching multigraph is useful for reasoning with plane graphs that are not triconnected. Adding this notion of latching graphs to the toolbox for triconnected planar graphs is one of our contributions.

The proofs of propositions and lemmas marked with $\star$ will be found in the journal version of this paper.

2 Preliminaries

Graphs. In this paper, all graphs of main interest are simple, that is, without self loops or parallel edges. Let G be a graph. We denote by $V(G)$ the vertex set of G and by $E(G)$ the edge set of G. When the vertex set of G is V, we say that the graph G is *on* V. The *complete graph* on V, denoted by $K(V)$, is a graph on V where every vertex is adjacent to all other vertices. A complete graph $K(V)$ is a K_n if $|V| = n$.

A graph H is a *subgraph* of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. The subgraph of G induced by $U \subseteq V(G)$ is denoted by $G[U]$: its vertex set is U and its edge set is $\{\{u,v\} \in E(G) \mid u,v \in U\}$. A vertex set $C \subseteq V(G)$ is a *clique* of G if $G[C]$ is a complete graph. For each $v \in V(G)$, $N_G(v)$ denotes the set of neighbors of v in G: $N_G(v) = \{u \in V(G) \mid \{u,v\} \in E(G)\}$. The *degree* of v in G is $|N_G(v)|$. For $U \subseteq V(G)$, the *open neighborhood of* U *in* G, denoted by $N_G(U)$, is the set of vertices adjacent to some vertex in U but not belonging to U itself: $N_G(U) = (\bigcup_{v \in U} N_G(v)) \setminus U$. For an edge set $E \subseteq E(G)$ we denote by $V(E)$ the set of vertices of G incident to some edge in E.

We say that a vertex set $C \subseteq V(G)$ is *connected in* G if, for every $u,v \in C$, there is a walk of $G[C]$ starting with u and ending with v, where a *walk* of G is a sequence of vertices in which every vertex except for the last is adjacent to the next vertex in the sequence. It is a *connected component* or simply a *component* of G if it is connected and is inclusion-wise maximal subject to connectivity. For each vertex set S of G, we denote by $\mathcal{C}_G(S)$ the set of components of $G[V(G) \setminus S]$. When G is clear from the context we may drop the subscript and write $\mathcal{C}(S)$ for $\mathcal{C}_G(S)$. A vertex set $S \subseteq V(G)$ is a *separator* of G if $|\mathcal{C}_G(S)| \geq 2$. We usually assume that G is connected and therefore $\emptyset$ is not a separator of G. A component $C \in \mathcal{C}_G(S)$ is a *full component* associated with S if $N_G(C) = S$. A graph G is *biconnected* (*triconnected*) if every separator of G is of cardinality two (three) or greater. A graph is a *cycle* if it is connected and every vertex is adjacent to exactly two vertices. A graph is a *tree* if it is connected and does not contain a cycle as a subgraph. A tree is a *path* if it has exactly two vertices of degree one or its vertex set is a singleton. Those two vertices of degree one are called the *ends* of the path. If p is a path with ends a and b, we say that p is *between* a and b. When we speak of cycles or paths of G, we mean subgraphs of G that are cycles or paths. Let p be a cycle or path of G. A *chord* of p in G is an edge $\{u,v\}$ of G with $u,v \in V(p)$ that is not an edge of p. A cycle or path of G is *chordless* if it does not have any chord in G.

Tree decompositions. A *tree-decomposition* of G is a pair $(T, \mathcal{X})$ where T is a tree and $\mathcal{X}$ is a family $\{X_i\}_{i \in V(T)}$ of vertex sets of G, indexed by the nodes of T, such that the following three conditions are satisfied. We call each X_i the *bag* at node i.

1. $\bigcup_{i \in V(T)} X_i = V(G)$.
2. For each edge $\{u, v\} \in E(G)$, there is some $i \in V(T)$ such that $u, v \in X_i$.
3. For each $v \in V(G)$, the set of nodes $I_v = \{i \in V(T) \mid v \in X_i\} \subseteq V(T)$ is connected in T.

The *width* of this tree-decomposition is $\max_{i \in V(T)} |X_i| - 1$. The *treewidth* of G, denoted by $tw(G)$ is the smallest k such that there is a tree-decomposition of G of width k.

Minimal separators and potential maximal cliques. Let G be a graph and S a separator of G. For distinct vertices $a, b \in V(G)$, S is an *a-b separator* if there is no path between a and b in $G[V(G) \setminus S]$; it is a *minimal a-b separator* if it is an a-b separator and no proper subset of S is an a-b separator. A separator is a *minimal separator* if it is a minimal a-b separator for some $a, b \in V(G)$. A necessary and sufficient condition for a separator S of G to be minimal is that $\mathcal{C}_G(S)$ has at least two members that are full components associated with S. We say that a connected vertex set C of G is *minimally separated* if $N_G(C)$ is a minimal separator.

Graph H is *chordal* if every cycle of H with four or more vertices has a chord in H. H is a *triangulation of graph G* if it is chordal, $V(G) = V(H)$, and $E(G) \subseteq E(H)$. A triangulation H of G is *minimal* if there is no triangulation H' of G such that $E(H')$ is a proper subset of $E(H)$.

A vertex set X of G is a *potential maximal clique* (PMC for short) of G if there is some minimal triangulation H of G such that X is a maximal clique of H. We denote by $\Pi(G)$ the set of all PMCs of G. The following lemmas due to Bouchitté and Todinca [8] are essential in our reasoning about PMCs.

▶ **Lemma 1** (Bouchitté and Todinca [7]). *Let X be a PMC of a graph G. Then, for every component C of $G[V(G) \setminus X]$, $S = N_G[C]$ has a full component containing $X \setminus S$ and hence is a minimal separator. Moreover, for every minimal separator S of G such that $S \subseteq X$, we have $S \neq X$ and there is some component C of $G[V(G) \setminus X]$ such that $S = N_G(C)$.*

▶ **Lemma 2** (Bouchitté and Todinca [7]). *A vertex set X of graph G is a PMC of G if and only if all of the following conditions hold.*

1. *There is no full component associated with X.*
2. *For every $u, v \in X$, either $\{u, v\} \in E(G)$ or there is some $C \in \mathcal{C}_G(X)$ such that $u, v \in N_G(C)$.*

They also show that PMCs are extremely useful for computing the treewidth.

▶ **Theorem 3** (Bouchitté and Todinca [7]). *Given a graph G of n vertices and $\Pi(G)$, the treewidth of G can be computed in time $|\Pi(G)| n^{O(1)}$.*

We need the following property of PMCs.

▶ **Lemma 4** (Bouchitté and Todinca [7]). *Let X be a PMC of a graph G. Then, every proper subset of X has a full component associated with it. As a consequence, no proper subset of a PMC is a PMC.*

Plane graphs. A *sphere-embedded graph* is a graph where a vertex is a point on the sphere Σ and each edge is a simple curve on Σ between two distinct vertices that does not contain any vertex except for its ends. In the combinatorial view of a sphere-embedded graph G, an edge represents the adjacency between two vertices at its ends. A sphere-embedded graph G is a *plane graph* if no two edges of G intersect each other as curves except at their ends. A graph G is *planar* if there is a plane graph that is isomorphic to G in the combinatorial view. We call this plane graph a *planar embedding* of G.

Let G be a biconnected plane graph. Then the removal of all the edges of G from the sphere Σ results in a collection of regions homeomorphic to open discs. We call these regions the *faces* of G. Each face of G is bounded by a simple closed curve that consists of edges of G. A vertex or an edge of G is *incident* to a face f of G, if it is contained in the closed curve that bounds f. We denote by $F(G)$ the set of faces of G. For each $f \in F(G)$, we denote by $V(f)$ the set of vertices of G incident to f.

A *combinatorial* representation of a plane graph G is an indexed set $\{\pi_v\}_{v \in V(G)}$, where π_v is a permutation on $N_G(v)$ such that $\pi_v(u)$ for $u \in N_G(v)$ is the neighbor of v that comes immediately after u in the clockwise order around v. Suppose we have two planar embeddings G_1 and G_2 of a graph G and assume that $V(G_1) = V(G_2) = V(G)$ and moreover the isomorphisms between G_1 and G as well as between G_2 and G are identities. Let, for $i \in \{1, 2\}$, the combinatorial representation of G_i be $\{\pi_{i,v}\}_{v \in V(G)}$. We say that these two planar embeddings are *combinatorially equivalent* if either $\pi_{1,v} = \pi_{2,v}$ for every $v \in V(G)$ or $\pi_{1,v} = (\pi_{2,v})^{-1}$ for every $v \in V(G)$. In this paper, we mainly deal with triconnected planar graphs. It is known that, for a triconnected planar graph G, the planar embedding of G is unique up to combinatorial equivalence [22]. For this reason, we will be dealing with triconnected plane graphs rather than triconnected planar graphs.

In conventional approaches for branch-decompositions and tree-decompositions of plane graphs [19, 6], the standard tools are radial graphs and nooses. We observe that, for triconnected plane graphs, latching graphs as defined below may replace those tools and greatly simplify definitions and reasoning.

▶ **Definition 5.** *Let G be a biconnected plane graph. The* latching graph *of G, denoted by L_G, is a multigraph obtained from G by, for each face f bounded by a cycle of four or more vertices, drawing every chord of the cycle within f.*

Figure 1 shows a triconnected plane graph with its latching graph and a plane graph for which the latching graph is not simple.

▶ **Proposition 6.** *The latching graph of a plane graph G is simple if G is triconnected.*

Proof. Let G be a triconnected plane graph and let u and v be two vertices of G. It is straightforward to see, and is observed in [6], that there are exactly two faces of G to which both u and v are incident if $\{u, v\}$ is an edge of G and there is at most one such face otherwise. In the first case, there is no face in which $\{u, v\}$ is a chord and, in the second case, there is at most one face in which $\{u, v\}$ is a chord. Thus, there is at most one edge between u and v in L_G and therefore L_G is simple. ◀

Although the latching graph L_G of a triconnected plane graph G is not a plane graph in general, we are interested in its subgraphs that are plane graphs. For $X \subseteq V(G)$, the *subgraph of L_G induced by X*, denoted by $L_G[X]$, is a sphere-embedded graph with vertex set X that inherits all edges of L_G, as curves, with two ends in X.

▶ **Proposition 7.** *Let G be a triconnected plane graph and let X be a vertex set of G. Then, $L_G[X]$ is a plane graph if and only if there is no face f of G such that $|V(f) \cap X| \geq 4$.*

Proof. Since $V(f) \cap X$ forms a clique of $L_G[X]$ for each face f of G, $L_G[X]$ is not a plane graph if there is some face f such that $|V(f) \cap X| \geq 4$. Conversely, if $L_G[X]$ has an edge-crossing, it must be in some face f of G and we have $|V(f) \cap X| \geq 4$. ◀

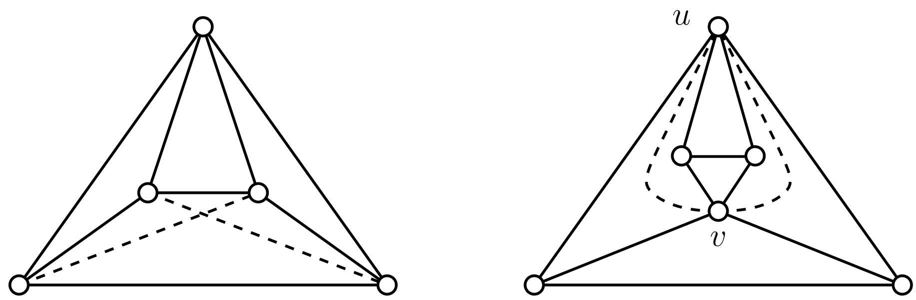

Figure 1 (A) a triconnected plane graph (solid edges) and its latching graph (solid and broken edges). (B) edge $\{u, v\}$ is drawn in more than one face if u and v separate the graph.

Let $X \subseteq V(G)$ such that $L_G[X]$ is a biconnected plane graph. We call each face of $L_G[X]$ a *region* of $L_G[X]$, in order to avoid confusions with the faces of G. We say that a region r of $L_G[X]$ is *empty* if r does not contain any vertex of G.

▶ **Proposition 8** ($\star$). *Let G be a triconnected plane graph and let $X \subseteq V(G)$ be such that $L_G[X]$ is a biconnected plane graph. Then, each $C \in \mathcal{C}_G(X)$ is contained in some region of $L_G[X]$. Moreover, each region of $L_G[X]$ contains at most one component in $\mathcal{C}_G(X)$ and, if the region is incident to four or more vertices, exactly one component.*

We use this framework to formulate the fundamental observation on minimal separators of triconnected plane graphs, which is conventionally stated in terms of nooses [6].

▶ **Proposition 9** ($\star$). *Let G be a triconnected plane graph. A vertex set S of G is a minimal separator of G if and only if $L_G[S]$ is a cycle and neither of the two regions of $L_G[S]$ is empty.*

▶ Remark 10. It follows from this proposition that every minimal separator of a triconnected planar graph has exactly two full components associated with it.

▶ **Corollary 11** ($\star$). *Let G be a triconnected plane graph. A vertex set S of G with $|S| \geq 4$ is a minimal separator of G if and only if $L_G[S]$ is a cycle.*

The following fact is essential in our treatment of PMCs of triconnected plane graphs.

▶ **Proposition 12** ($\star$). *For every PMC X of a triconnected plane graph G, $L_G[X]$ is a biconnected plane graph. Moreover, every region of $L_G[X]$ is bounded by a chordless cycle of L_G.*

3 Characterizing PMCs in triconnected planar graphs

In this section, we characterize PMCs of a triconnected plane graph in terms of graphs we call steerings.

▶ **Definition 13.** *Let γ be a cycle. A subset R of $V(\gamma)$ is a* slot *of γ if R is a singleton or an edge of γ. A graph H is a* steering, *if there is a bipartition (S, P) of $V(H)$ such that $H[S]$ is a cycle, $N_H(P)$ is neither empty nor a slot of $H[S]$, and if $|P| \geq 2$ then the following conditions hold.*

1. *$H[P]$ is a path.*
2. *No internal vertex of the path $H[P]$ is adjacent to any vertex in S.*
3. *For each end t of the path, $N_H(t) \cap S$ is a slot of $H[S]$.*

*We call H an (S, P)-*steering *in this situation. We call a steering a* wheel *if it is an (S, P)-steering for some bipartition (S, P) of $V(H)$ such that $|P| = 1$; otherwise it is a* non-wheel steering.

Figure 2 shows some steerings. Figure 3 shows two steerings with different (S, P)-bipartitions. The example (B) is an (S, P)-steering for the second bipartition but not for the first bipartition. Figure 4 shows some graphs that are not steerings.

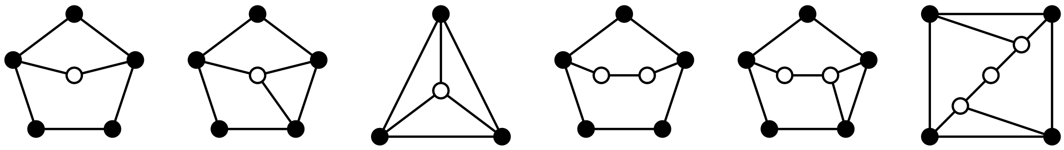

Figure 2 Some steerings. White vertices belong to P and black vertices belong to S, in a choice of the bipartition (S, P) that works. The first three are wheels while the remaining three are non-wheels.

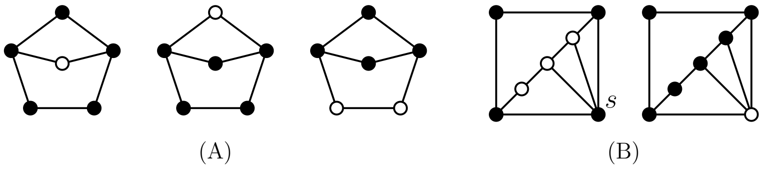

Figure 3 (A) A steering with three possible (S, P) bipartitions. Because of the first or the second bipartition, it is a wheel. (B) Also a wheel. For the first (S, P) bipartition, it is not an (S, P)-steering since an internal vertex of the path on P are adjacent to $s \in S$. The second (S, P) bipartition shows that it is a wheel.

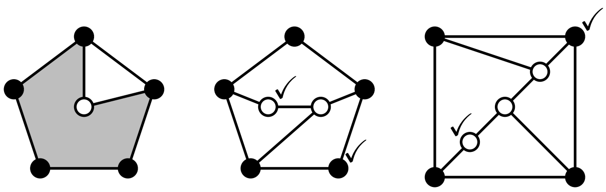

Figure 4 Some graphs that are not steerings. In the shown (S, P) bipartition, the first example H is not a steering since $N_H(P) \cap S$ is a slot. The second example is not a steering since $N_H(t) \cap S$ is not a slot for the right end t of the path $H[P]$. The third example is not a steering since an internal vertex of the path $H[P]$ is adjacent to a vertex in S. It can be verified that these graphs are not (S, P)-steerings for any other (S, P) bipartition. It is explained in the main text why any of these plane graphs cannot be $L_G[X]$ for a PMC X of a triconnected plane graph G.

▶ **Proposition 14** ($\star$). *Let H be a steering. Then H is biconnected and planar. Moreover, the planar embedding of H is unique up to combinatorial equivalence.*

From now on, when we refer to a steering H, we view H as a plane graph rather than a combinatorial graph as originally defined.

The following lemma shows that X is a PMC of a triconnected plane graph G if $L_G[X]$ is a steering. Before going into technical details, it may be helpful to study the examples in Figures 2 and 3 and confirm the following.

1. The cycle bounding each face is chordless.
2. Every pair of vertices is either adjacent to each other or incident to a common face.

Also observe that if $L_G[X]$ is one of the non-steering graphs in Figure 4, then X cannot be a PMC of G: in the first example, the component of $G[V(G) \setminus X]$ lying in the gray face would be a full component of X, violating the first condition of Lemma 2 for X to be a PMC of G; in the second and third examples, the pair of checked vertices do not share a common face and therefore cannot belong to the neighborhood of a common component of $G[V(G) \setminus X]$, violating the second condition of Lemma 2 for X to be a PMC of G.

► **Lemma 15.** *Let X be a vertex set of a triconnected plane graph G and suppose $L_G[X]$ is a steering. Then, X is a PMC of G.*

Proof. Suppose $L_G[X] = H$ where H is an (S, P)-steering. This equality means an equality as plane graphs, since we are viewing a steering as a biconnected plane graph, as we mentioned in the remark following Proposition 14.

Due to Proposition 8, each component in $\mathcal{C}_G(X)$ is contained in a region of H and each region of H contains at most one component in $\mathcal{C}_G(X)$.

We show that X satisfies the two conditions for PMCs in Lemma 2: (1) There is no full component associated with X; (2) For every pair of vertices x and y in X, either x is adjacent to y in G or there is some $C \in \mathcal{C}_G(X)$ such that $x, y \in N_G(C)$.

To show that Condition (1) holds, let C be an arbitrary component in $\mathcal{C}_G(X)$ and γ_C be the cycle bounding the region of H containing C. If γ_C is $H[S]$, then C is not a full component associated with X as S is a proper subset of X. Otherwise, there is a vertex in S that does not belong to γ_C, since $N_H(P)$ is not a slot of the cycle $H[S]$, and therefore C is not a full component associated with X.

To show that Condition (2) holds, we first show that, for arbitrary two members x and y of X, there is some region of H whose boundary contains both x and y. If $x, y \in S$, then this certainly holds since H has a region bounded by the cycle $H[S]$. So suppose that at least one of x and y, say x, belongs to P. We argue in a few cases.

First suppose that $|P| = 1$ and hence $P = \{x\}$ and $y \in S$. Then, there is certainly a region of H bounded by a cycle consisting of x and a subpath of the cycle $H[S]$ that contains y. See the first three examples in Figure 2.

Next suppose that $|P| \geq 2$. Since there is no internal vertex of the path $H[P]$ adjacent in H to any vertex in S, there are two regions of H bounded by cycles containing the path $H[P]$. Let γ_1 and γ_2 be those cycles. Then, γ_i, for each $i \in \{1, 2\}$, consists of $H[P]$ and a subpath p_i of $H[S]$. Since $N_H(t) \cap S$ for each end t of the path $H[P]$ is a slot, we have $V(p_1) \cap V(p_2) = S$. Therefore, either γ_1 or γ_2 contains both x and y. See the last three examples in Figure 2.

We are ready to show that Condition (2) holds. Let x and y be arbitrary two vertices in X. As we have shown above, there is a cycle γ bounding a region of H such that $x, y \in V(\gamma)$. If this region contains some $C \in \mathcal{C}_G(X)$, then we are done since $N_G(C) = V(\gamma)$. Otherwise, γ bounds an empty region and must be a triangle since otherwise $L_G[X]$ would not be a plane graph. Therefore, $\{x, y\}$ is an edge of $L_G[X]$. If this is an edge of G then we are done, so suppose not. Let γ' be another cycle bounding a region of $L_G[X]$ that contains the edge $\{x, y\}$. This region bounded by γ' cannot be empty since, if it were, then we would have a face of G incident to four or more vertices of X, and $L_G[X]$ would not be a plane graph. We are done, since $x, y \in N_G(C')$ where C' is the member of $\mathcal{C}_G(X)$ contained in the region bounded by γ'. ◄

The converse of this lemma is shown in several steps. We start with a technical proposition.

▶ **Proposition 16.** *Let X be a PMC of a triconnected plane graph G. If $L_G[X']$ is a steering for some $X' \subseteq X$, then $X = X'$ and hence $L_G[X]$ is a steering.*

Proof. If $L_G[X']$ is a steering, X' is a PMC of G due to Lemma 15. Since no proper subset of a PMC is a PMC (Lemma 4), we have $X = X'$. ◀

The next lemma deals with a special case where every minimal separator contained in a PMC X consists of three vertices.

▶ **Lemma 17.** *Let X be a PMC of a triconnected plane graph G and suppose that, for every $C \in \mathcal{C}_G(X)$, $|N_G(C)| = 3$. Then, $L_G[X]$ is a K_4.*

Proof. Recall that, due to Proposition 12, $L_G[X]$ is a biconnected plane graph. Recall also that every $C \in \mathcal{C}_G(X)$ is contained in a region of $L_G[X]$ bounded by a cycle whose vertex set is $N_G(C)$. Let γ be a cycle that bounds a region r of $L_G[X]$ and let $S = V(\gamma)$. If r is empty then $|S| = 3$ since otherwise $L_G[X]$ would not be a plane graph (Proposition 7). Otherwise r contains some $C \in \mathcal{C}_G(X)$ and we also have $|S| = 3$ due to the assumption. Let $x \in X \setminus S$ and $s \in S$ be arbitrary. We show that x is adjacent to s in $L_G[X]$. If x is adjacent to s in G then we are done. Otherwise, due to the second condition for PMCs in Lemma 2, there is some $C \in \mathcal{C}_G(X)$ such that $x, s \in N_G(C)$. Since the region of $L_G[X]$ containing C is bounded by a triangle on $N_G(C)$, x is adjacent to s in $L_G[X]$. Therefore, x is adjacent in $L_G[X]$ to every $s \in S$ and hence $L_G[S \cup \{x\}]$ is a K_4. As $L_G[S \cup \{x\}]$ is an $(S, \{x\})$-steering, we have $X = S \cup \{x\}$ due to Proposition 16 and hence $L_G[X]$ is a K_4. ◀

To deal with the more general case, we use the following notion of *arches*.

▶ **Definition 18.** *Let G be a triconnected plane graph and let S be a minimal separator of G with $|S| \geq 4$. An* arch *of S is a subset P of $V(G) \setminus S$ such that $L_G[P]$ is a path and $N_{L_G}(P) \cap S$ is neither empty nor a slot of the cycle $L_G[S]$.*

If $L_G[S \cup P]$ is an (S, P)-steering, then P is certainly an arch of S. The converse does not hold: if P is an arch of S then $L_G[S \cup P]$ is not necessarily an (S, P)-steering, as the definition of steerings requires more conditions to be satisfied by $L_G[S \cup P]$. We show, however, in the next lemma that if P is an inclusion-wise minimal arch then $L_G[S \cup P]$ is a steering. A caveat: $L_G[S \cup P]$ is not necessarily an (S, P)-steering in this case; it may be an $(S', \{s\})$-steering where $S' = (S \cup P) \setminus \{s\}$ for some $s \in S$. In Lemma 20, we show that if X is a PMC then every minimal separator $S \subset X$ with $|S| \geq 4$ has an arch P that is a subset of $X \setminus S$. It follows from these two lemmas that, for every minimal separator S with $|S| \geq 4$ that is a subset of a PMC X, there is some $P \subset X \setminus S$ such that $L_G[S \cup P]$ is a steering. Due to Lemma 15, $S \cup P$ is a PMC and, since no proper subset of a PMC is a PMC (Lemma 4), we conclude that $X = S \cup P$ and $L_G[X]$ is a PMC, which is an essential part of Theorem 21.

▶ **Lemma 19.** *Let S be a minimal separator of a triconnected plane graph G with $|S| \geq 4$. Suppose S has an arch and let P be an inclusion-wise minimal arch of S. Let $H = L_G[S \cup P]$. Then, either H is an (S, P)-steering or there is some $s \in N_{L_G}(P) \cap S$ such that H is an $((S \cup P) \setminus \{s\}, \{s\})$-steering.*

Proof. If there is some $v \in P$ such that $N_{L_G}(v) \cap S$ is neither empty nor a slot of the cycle $L_G[S]$, then $\{v\}$ is an arch of S and, since P is minimal, $P = \{v\}$. We are done, since H is an $(S, \{v\})$-steering.

Suppose $N_{L_G}(v) \cap S$ is either empty or a slot of $L_G[S]$ for every $v \in P$. Since $N_{L_G}(P) \cap S$ is not a slot of $L_G[S]$, there are two vertices $v_1, v_2 \in P$ and two vertices $s_1, s_2 \in S$ such that v_i is adjacent to s_i for $i \in \{1, 2\}$ and $\{s_1, s_2\}$ is not an edge of S. Let p be a shortest path in $L_G[P]$ between v_1 and v_2. Then, $V(p)$ is an arch of S and, since P is minimal, P must be equal to $V(p)$. If no internal vertex of p is adjacent in L_G to any vertex in S, then $L_G[S \cup P]$ is an (S, P)-steering and we are done. So suppose an internal vertex v of p is adjacent to some vertex in S. Let p_i be the subpath of p between v_i and v, for $i \in \{1, 2\}$. Since P is a minimal arch of S, neither $V(p_1)$ nor $V(p_2)$ is an arch of S. Therefore, $N_{L_G}(\{v_i, v\}) \cap S$ is a slot for $i \in \{1, 2\}$. This is possible only if there is a vertex $s \in S$ such that s is adjacent to both s_1 and s_2 on $L_G[S]$, $N_{L_G}(v_i)$ is either $\{s_i\}$ or $\{s_i, s\}$ for $i \in \{1, 2\}$, and $N_{L_G}(v) \cap S = \{s\}$. Moreover, since $P = V(p)$ is a minimal arch of S, no vertices in P other than v, v_1, or v_2 are adjacent to any vertex in S. Let p_3 be the subpath of $L_G[S]$ between s_1 and s_2 avoiding s and let γ be the cycle consisting of p_3 and p. Let $S' = V(\gamma) = (S \cup P) \setminus \{s\}$. We claim that γ is chordless in L_G, that is, $L_G[S'] = \gamma$. This is because p_3 is chordless since $L_G[S]$ is a cycle by assumption, p is chordless since it is the shortest path in P between v_1 and v_2, and the only edges between $V(p)$ and $V(p_3)$ are the edges $\{v_1, s_1\}$ and $\{v_2, s_2\}$. Since $N_{L_G}(s) \cap S' = \{s_1, v, s_2\}$, $L_G[S \cup P]$ is an $(S', \{s\})$-steering as desired. ◀

▶ **Lemma 20.** *Let X be a PMC of a triconnected plane graph G and let $S \subseteq X$ be a minimal separator of G such that $|S| \geq 4$. Then, there is an arch P of S such that $P \subseteq X \setminus S$.*

Proof. Due to Lemma 1, $X \setminus S$ is non-empty and is a subset of one of the two full components associated with S. Let r_0 be the region of $L_G[S]$ that contains $X \setminus S$.

Suppose that there is no arch P of S such that $P \subseteq X \setminus S$. Then, for every pair of vertices s and s' in S that are not adjacent to each other on the cycle $L_G[S]$, we can draw a curve between s and s' in r_0 without crossing the drawing of $L_G[X]$. Therefore, there is a region r of $L_G[X]$ contained in r_0 that is incident to all vertices in S. Let γ be the cycle of $L_G[X]$ bounding r. Since $|V(\gamma)| \geq |S| \geq 4$, r is non-empty (Proposition 8) and hence contains a component $C \in \mathcal{C}_G(X)$. Therefore, $V(\gamma)$ must be a minimal separator of G, due to Lemma 1. But this is impossible, since some edge of $L_G[S]$ is a chord of γ, contradicting Proposition 9. ◀

The following is our main theorem in this section.

▶ **Theorem 21.** *A vertex set X of a triconnected plane graph G is a PMC if and only if $L_G[X]$ is a steering.*

Proof. Lemma 15 shows that if $L_G[X]$ is a steering then X is a PMC. We prove the other direction. Let X be a PMC of G.

If $|N_G(C)| = 3$ for every $C \in \mathcal{C}_G(X)$, $L_G[X]$ is a K_4 due to Lemma 17 and hence is an $(X \setminus \{x\}, \{x\})$-steering for every $x \in X$.

Suppose there is some $C \in \mathcal{C}_G(X)$ with $|S| \geq 4$ where $S = N_G(C)$. Due to Lemma 20, there is an arch of S that is a subset of $X \setminus S$. Let P be an inclusion-wise minimal arch of S that is a subset of $X \setminus S$. Due to Lemma 19, $L_G[S \cup P]$ is a steering and hence $S \cup P$ is a PMC of G, due to Lemma 15. Due to Proposition 16, $X = S \cup P$ and hence $L_G[X]$ is a steering. ◀

4 Polynomial delay generation of PMCs of a triconnected planar graph

Let J be some combinatorial structure of size n and let $\mathcal{S}(J)$ be the set of some combinatorial objects defined on J. We say an algorithm *generates* $\mathcal{S}(J)$ *with polynomial delay* if it outputs each member of $\mathcal{S}(J)$ exactly once and the time between two consecutive events is $n^{O(1)}$, where an event is the initiation of the algorithm, an output, or the termination of the algorithm.

The basic approach for generating $\Pi(G)$ for a given triconnected plane graph G is as follows. PMCs X such that $|N_G(C)| = 3$ for every $C \in \mathcal{C}_G(X)$ can be trivially generated with polynomial delay, since $L_G[X]$ is a K_4 for each such X due to Lemma 17. So, we concentrate on generating $\Pi'(G)$ consisting of members X of $\Pi(G)$ such that $\mathcal{C}_G(X)$ contains a component C with $|S| \geq 4$ where $S = N_G(C)$.

We need an algorithm to generate minimal separators of G. Although a polynomial delay algorithm for this task is known for general graphs [2], we use an algorithm specialized for triconnected plane graphs for a technical reason to become clear below.

We need the following result due to Uno and Satoh [21].

▶ **Lemma 22** (Uno and Satoh [21]). *Given a graph G, all chordless cycles of G can be generated with polynomial delay. Given a graph G and two vertices $s, t \in V(G)$, all chordless paths of G between s and t can be generated with polynomial delay.*

▶ **Lemma 23.** *Given a triconnected plane graph G, all minimal separators of G can be generated with polynomial delay. Moreover, for each $v \in V(G)$, all minimal separators of G that do not contain v can be generated with polynomial delay.*

Proof. Given G, we generate all chordless cycles of L_G with polynomial delay, using the algorithm of Uno and Satoh. These chordless cycles are in one-to-one correspondence with minimal separators of G due to Proposition 9. For the second part, we generate all chordless cycles of $G[V(G) \setminus \{v\}]$. Since a chordless cycle of $G[V(G) \setminus \{v\}]$ is a chordless cycle of G, we have the result. ◀

▶ Remark 24. It is not clear if the algorithm [2] for generating minimal separators of a general graph can be adapted to support the second part of the lemma, which is used in the proof of Theorem 32. We also note that the chordless path part of Lemma 22 is used in the proof of Lemma 30.

To design an algorithm to generate all PMCs based on our characterization of PMCs stated in Theorem 21, we need some preparations.

▶ **Definition 25.** *Let G be a triconnected plane graph and C a minimally separated component of G with $|S| \geq 4$ where $S = N_G(C)$. Recall that there are exactly two full components associated with S and let C' be the full component distinct from C. A* port *of (S, C) is a vertex $u \in C'$ such that $N_{L_G}(u) \cap S$ is a slot of $L_G[S]$. We call the slot $N_{L_G}(u) \cap S$ of $L_G[S]$ the* slot *for u. A pair of ports u_1 and u_2 is* valid *if the union of the slot for u_1 and the slot for u_2 is not a slot. A vertex $s \in S$ with two neighbors s_1 and s_2 on the cycle $L_G[S]$ is a* hinge *of a valid pair of ports u_1 and u_2 if the slot for u_1 is either $\{s_1\}$ or $\{s_1, s\}$ and the slot for u_2 is either $\{s_2\}$ or $\{s_2, s\}$. See Figure 5 for examples of ports, valid and invalid port pairs, and a hinge.*

Ports are potential ends of a path p such that $L_G[X]$ for $X = S \cup V(p)$ is a steering. A pair of ports must be valid in order for the pair to be the ends of such path p. We generate such paths using the algorithm for generating chordless paths. The following lemma is the basis of this approach.

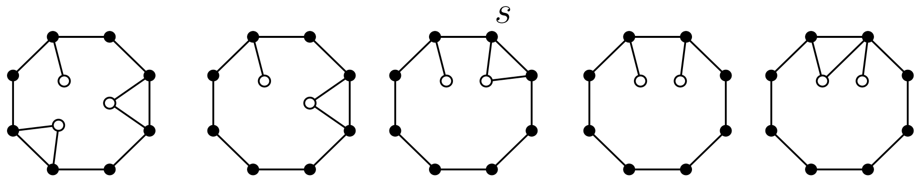

Figure 5 (A) White vertices are some ports of (S, C) where S consists of black vertices and C is the full component associated with S that lies in the outer face of the black cycle. (B) Some valid pairs of ports. The valid pair in the second example has a hinge s. (C) Some invalid pairs of ports.

▶ **Lemma 26.** *Let G be a triconnected plane graph and C a minimally separated component of G with $|S| \geq 4$, where $S = N_G(C)$. Let C' be the other full component of S. Let P be an arbitrary subset of C' with $|P| \geq 2$ and let $X = S \cup P$.*

1. *For each valid pair u_1 and u_2 of ports of (S, C), let $A(C, u_1, u_2)$ be the subgraph of L_G induced by the vertex set $(C' \setminus N_{L_G}(S)) \cup \{u_1, u_2\}$. Then, $L_G[X]$ is an (S, P)-steering if and only if there is a valid pair u_1 and u_2 of the ports of S such that $P = V(p)$ for some chordless path p of $A(C, u_1, u_2)$ between u_1 and u_2.*
2. *For each valid pair u_1 and u_2 of ports of (S, C) that has hinges and each hinge s of the pair, let $B(C, u_1, u_2, s)$ be the subgraph of L_G induced by the vertex set $(C' \setminus N_{L_G}(S)) \cup N_{L_G}(s) \cup \{u_1, u_2\}$. Then, $L_G[X]$ is an $(S', \{s\})$-steering for $s \in S$, where $S' = X \setminus \{s\}$, if and only if there is a valid pair of ports u_1 and u_2 of (S, C) with a hinge s such that $P = V(p)$ for some chordless path p of $B(C, u_1, u_2, s)$ between u_1 and u_2.*

Proof. For part 1, suppose $L_G[X]$ is an (S, P)-steering where $P \subseteq C'$. Let $H = L_G[X]$. Due to Definition 13, $H[P] = L_G[P]$ is a path. Let u_1 and u_2 be the two ends of this path. Then, u_1 and u_2 are ports of S and, moreover, the pair of these ports is valid. Since no internal vertex of the path $L_G[P]$ is adjacent in L_G to any vertex in S, $L_G[P]$ is a path in $A(C, u_1, u_2)$. It is a chordless path, since if it had a chord in $A(C, u_1, u_2)$, then the graph $L_G[P]$ would contain that chord and $L_G[P]$ would not be a path.

For the other direction, suppose that there is a valid pair of ports u_1 and u_2 of S such that $L_G[P]$ is a chordless path of $A(C, u_1, u_2)$ between u_1 and u_2. Then, $N_{L_G}(P \setminus \{u_1, u_2\}) \cap S$ is empty and hence $L_G[S \cup P]$ is an (S, P)-steering.

For part 2, suppose $L_G[X]$ is an $(S', \{s\})$-steering where $s \in S$ and $S' = X \setminus \{s\}$. Due to the proof of Lemma 19, $L_G[P]$ is a path of L_G between some valid pair of ports u_1 and u_2 of (S, C) and, moreover, s is a hinge of this pair of ports. Since the only vertex in S that can be adjacent to any internal vertex of $L_G[P]$ is s, $L_G[P]$ is a path in $B(C, u_1, u_2, s)$ between u_1 and u_2. It is a chordless path, since if it had a chord in $B(C, u_1, u_2, s)$, then the graph $L_G[P]$ would contain that chord and $L_G[P]$ would not be a path.

For the other direction, suppose that there is a valid pair of ports u_1 and u_2 of S with a hinge s such that $L_G[P]$ is a chordless path of $B(C, u_1, u_2, s)$ between u_1 and u_2. Then, $L_G[S']$, where $S' = X \setminus \{s\}$, is a cycle and $L_G(X)$ is an $(S', \{s\})$-steering. ◀

In implementing the generation of PMCs based on this lemma, we need to deal with the problem of suppressing duplicate outputs of a single element without introducing super-polynomial delay. We use the technique due to Bergougnoux, Kanté and Wasa [1] to address this problem. They used this technique for a particular problem of generating what they call minimal disjunctive separators. We formulate their technique in the following theorem that can be used for wide range of generation problems.

Algorithm 1 Generation of $\mathcal{S}(G)$, using sub-generators Gen_i for $\mathcal{S}_i(G)$, $i \in I$.

▷ We say *emit* s to mean that this algorithm outputs s, in order to distinguish this action from the output events of the sub-generators

▷ We say that an output of a sub-generator is *suppressed* if it is not emitted

Ensure: Generate all members of $\mathcal{S}(G)$

```
for each i ∈ I do
    Initiate Gen_i and immediately suspend it
end for
i* ← 1
repeat
    ascending ← true
    while ascending do
        Execute Gen_{i*} up to the next event
        if the event is the termination event then
            ascending ← false
        else                                          ▷ the event is an output event
            Let s be the object to be output
            if s ∈ S_i(G) for some i > i* then        ▷ the output of s is suppressed
                i* ← the smallest j > i* such that Gen_j has not been terminated
                                                      ▷ such j exists, see the proof
            else
                Emit s
                ascending ← false
            end if
        end if
    end while
    if there is some j such that Gen_j has not been terminated then
        i* ← the smallest j such that Gen_j has not been terminated
    end if
until Gen_i has been terminated for every i ∈ I
```

Let G be a graph on n vertices. We consider the general task of generating some structures defined on G. Let $\mathcal{S}(G)$ denote the set of those structures to be generated. Suppose N subsets $\mathcal{S}_1(G), \ldots, \mathcal{S}_N(G)$ of $\mathcal{S}(G)$ are defined. Let $I = \{i \mid 1 \leq i \leq N\}$. Suppose that the following conditions are satisfied.

1. $\mathcal{S}(G) = \bigcup_{i \in I} \mathcal{S}_i(G)$.
2. $N = n^{O(1)}$.
3. For each $s \in \mathcal{S}(G)$ and $i \in I$, it can be decided whether $s \in \mathcal{S}_i(G)$ in time $n^{O(1)}$.
4. For each $i \in I$, there is an algorithm Gen_i that generates $\mathcal{S}_i(G)$ with polynomial delay.

▶ **Theorem 27.** *Under the above assumptions, $\mathcal{S}(G)$ can be generated with polynomial delay.*

We prove this theorem by showing that Algorithm 1 generates $\mathcal{S}(G)$ with polynomial delay. The idea of this algorithm is as follows. For each $s \in \mathcal{S}(G)$, let us say that $i \in I$ *owns* s if i is the largest j such that $s \in \mathcal{S}_j(G)$. We let Gen_i, where i owns s, be responsible for the output of s and suppress the output of s from Gen_j for other j. The executions of Gen_i for $i \in I$ are scheduled in such a way that at most a polynomial number of outputs are suppressed before an unsuppressed output occurs.

▶ **Lemma 28** (⋆). *Algorithm 1 generates $\mathcal{S}(G)$ with polynomial delay, if all of the four conditions above are satisfied.*

We now turn to the application of this technique to our goal.

For a minimally separated component C of a triconnected plane graph G with $|S| \geq 4$ where $S = N_G(C)$, let $\Pi(G, C)$ denote the set of PMCs X such that $C \in \mathcal{C}_G(X)$ and $L_G[X]$ is a steering.

▶ **Proposition 29.** *Let G be a triconnected plane graph and C a minimally separated component of G such that $|N_G(C)| \geq 4$. Then, $\Pi(G, C) \neq \emptyset$.*

Proof. Let $S = N_G(C)$ and let C' be the other full component of S. Let H be an arbitrary minimal triangulation of G in which S is a clique. Let X be an arbitrary maximal clique of $H[S \cup C']$ that contains S. Since S is a separator of H, X is a maximal clique of H. Since no maximal clique of a chordal graph is a minimal separator, X is a proper superset of S and belongs to $\Pi(G, C)$. ◀

▶ **Lemma 30.** *Given a triconnected plane graph G and a minimally separated component C of G with $|S| \geq 4$ where $S = N_G(C)$, $\Pi(G, C)$ can be generated with polynomial delay.*

Proof. Based on Lemma 26, we generate $\Pi(G, C)$ as follows. We first generate $X \in \Pi(G, C)$ such that $L_G[X]$ is an (S, P)-steering where $P = X \setminus S$. To do so, for every valid pair of ports u_1 and u_2 of S, we generate all chordless paths of $A(C, u_1, u_2)$ between u_1 and u_2, where $A(C, u_1, u_2)$ is as defined in Lemma 26, using the algorithm due to Sato and Uno given in Lemma 22. For each path p generated, we output $S \cup V(p)$. Since the number of valid pairs of ports is $O(n^2)$, the generation of X in this category is with polynomial delay even if some valid pairs do not yield any such X.

We then generate $X \in \Pi(G, C)$ such that $L_G[X]$ is an $(S', \{s\})$-steering where $s \in S$ and $S' = X \setminus \{s\}$. To do so, for every valid pair of ports u_1 and u_2 of S with hinges and for each hinge s of this pair, we generate all chordless paths of $B(C, u_1, u_2, s)$ between u_1 and u_2. For each path p generated, we output $S \cup V(p)$. The valid pair of u_1 and u_2 may have more than one ports when $|S| = 4$, which could cause duplicate outputs of an identical X. We suppress such duplicate outputs using Theorem 27. Let $\mathcal{T}$ be the set of all triples (u_1, u_2, s) where (u_1, u_2) is a valid pair of ports of (S, C) and s is a hinge of this pair. Index the set $\mathcal{T}$ by $I = \{1, \ldots, N\}$ where $N = |\mathcal{T}|$. For each $i \in I$, let $\mathcal{P}_i$ denote the set of chordless paths of $B(C, u_1, u_2, s)$ between u_1 and u_2, where (u_1, u_2, s) is the triple in $\mathcal{T}$ indexed by i. Let $\mathcal{P} = \bigcup_{i \in I} \mathcal{P}_i$. Our goal is to generate $\mathcal{P}$ without duplication. We verify the conditions for Theorem 27 to be applied.

1. $\mathcal{P} = \bigcup_{i \in I} \mathcal{P}_i$ by definition.
2. $N = |I| = O(n^3)$.
3. For each path p in $\mathcal{P}$ and each $i \in I$, it can be decided whether $p \in \mathcal{P}_i$ in time $n^{O(1)}$.
4. For each $i \in I$, we can generate $\mathcal{P}_i$ with polynomial delay as described above.

Therefore, Theorem 27 applies and we can generate $\mathcal{P}$ and hence all PMCs X in this category, with polynomial delay. ◀

▶ Remark 31. Since the PMCs X in the second category in the above lemma, those such that $L_G[X]$ is an $(S', \{s\})$-steering for some $s \in S$ where $S' = X \setminus \{s\}$, also belong to $\Pi(G, C')$ for some full component associated with S', it might appear that the cumbersome generation of those PMCs described in the above proof is unnecessary. However, if we generate only the members of $\Pi(G, C)$ in the first category, those X such that $L_G[X]$ is an (S, P)-steering where $P = X \setminus S$, then it is possible for the generator of $\Pi(G, C)$ to produce no outputs. This is a problem, since there is no readily provable polynomial bound on the number of successive components C for which the generation of $\Pi(G, C)$ produces no outputs.

The following is the main result of this section.

▶ **Theorem 32.** *Given a triconnected plane graph G, $\Pi(G)$ can be generated with polynomial delay.*

Proof. It suffices to show that $\Pi'(G)$, the set of PMCs X of G such that $|N_G(C)| \geq 4$ for some $C \in \mathcal{C}_G(X)$, can be generated with polynomial delay. Let $n = |V(G)|$. For each $v \in G$, let $\Pi_v(G)$ denote the union of $\Pi(G, C)$ over all minimally separated components C of G such that $v \in C$ and $|N_G(C)| \geq 4$. We have $\Pi'(G) = \bigcup_{v \in V(G)} \Pi_v(G)$ and $|V(G)| = n = n^{O(1)}$. Given $X \in \Pi(G)$ and $v \in V(G)$, it is straightforward to test if $X \in \Pi_v(G)$ in polynomial time. For each v, $\Pi_v(G)$ can be generated with polynomial delay: we generate all minimal separators S such that $v \notin S$ by the second part of Lemma 23 and, for each S generated such that $|S| \geq 4$, generate $\Pi(G, C)$ for the full component C of S such that $v \in C$. The delay in this algorithm is polynomial, since minimal separators are generated with polynomial delay and, for each minimal separator generated, at least one PMC is generated due to Proposition 29.

Therefore, applying Theorem 27, we can generate $\Pi(G)$ with polynomial delay. ◀

5 Computing the planar treewidth

An algorithm for generating PMCs immediately leads to a treewidth algorithm for triconnected planar graphs because of Theorem 3 due to Bouchitté and Todinca [7]. To extend the algorithm for general planar graphs, we use the following result from [5]. A vertex set S of a graph G is an *almost clique* of G if there is some $v \in S$ such that $S \setminus \{v\}$ is a clique of G.

▶ **Theorem 33** (Bodlaender and Koster [5])**.** *Let G be a graph and S a minimal separator of G that is an almost clique. Let C_i, $1 \leq i \leq m$, be the components of $G[V(G) \setminus S]$ and let $G_i = G[C_i \cup N_G(C_i)] \cup K(N_G(C_i))$, for $i = 1, \dots, m$, be the graph obtained from $G[C_i \cup N_G(C_i)]$, the subgraph of G induced by the closed neighborhood of C_i, by filling the open neighborhood of C_i into a clique. Then $tw(G) = \max\{|S|, \max_{i \in \{1,\dots,m\}} tw(G_i)\}$.*

We use this theorem for the special case where S is a two-vertex minimal separator: such S is always an almost clique.

▶ **Theorem 34.** *Given a planar graph G, $tw(G)$ can be computed in time $|\Pi(G)|n^{O(1)}$.*

Proof. We assume that G is biconnected. Otherwise, we compute the treewidth of each biconnected component and take the maximum.

We first prove the case where G is triconnected. We compute a planar embedding of G in time $O(n)$ [15]. Then we apply our generation algorithm to compute $\Pi(G)$ in time $|\Pi(G)|n^{O(1)}$. Finally, we apply Theorem 3 to obtain $tw(G)$ in time $|\Pi(G)|n^{O(1)}$.

For a general planar graph G, let $s(G)$ denote the number of two-vertex separators of G. We prove the theorem by induction on $s(G)$. The base case $s(G) = 0$, where G is triconnected, is already proved. Suppose $s(G) > 0$ and let S be an arbitrary two-vertex separator of G. Let $C_1, \dots, C_m$ be the components of $G[V(G) \setminus S]$. Since G is biconnected, we have $N(C_i) = S$ for every $i \in \{1, \dots, m\}$. Therefore, S is a minimal separator of G and, due to Theorem 33, we have $tw(G) = \max\{2, \max_{i \in \{1,\dots,m\}} tw(G_i)\}$ where $G_i = G[C_i \cup S] \cup K(S)$. Since G_i is a subgraph of a minor of G obtained by contracting $C_{i'} \cup v$ into a single vertex for arbitrary $i' \neq i$ and $v \in S$, G_i is planar. Moreover, G_i is biconnected since any cut vertex of G_i would be a cut vertex of G. Therefore, we may apply the induction hypothesis to each G_i and compute $tw(G_i)$ for each i in time $|\Pi(G_i)|n^{O(1)}$. We claim that $\sum_i |\Pi(G_i)| \leq |\Pi(G)|$. To see this, let X be an arbitrary PMC of G_i and H a minimal triangulation of G_i in which X is a maximal clique. Let H' be an arbitrary minimal triangulation of G such that $H'[V(G_i)] = H$. Since every minimal separator of G that is a clique in a minimal triangulation of G is a minimal separator of that triangulation [14], S is a minimal separator of H'. Therefore, X is a maximal clique of H' and hence is a PMC of G. Moreover, X cannot be a PMC of G_j for any $j \neq i$ since X is not a subset of $V(G_j)$. Therefore, the claim holds. We conclude that we can compute $tw(G)$ in time $|\Pi(G)|n^{O(1)}$. ◀

6 Future work

Our success in a special case of the open question whether $\Pi(G)$ can be computed in time $|\Pi(G)|n^{O(1)}$ does not seem to suggest any approach to solving the question for general graphs. Much more modest goal is to address the question for general planar graphs. It might be possible to extend our generation algorithm to general planar graphs by finding some way of handling PMCs that cross two-vertex minimal separators. Bart Jansen asked if the generation result could be extended to graphs embedded on tori. We observe that latching graphs are not appropriate tool for that purpose, since if G is a triconnected graph embedded on a torus and X is a PMC of G then the subgraph of the latching graph of G, appropriately defined on tori, induced by X may have edge crossings: the proof of Proposition 12 relies essentially on the assumption that G is embedded in the sphere. Studying upper bounds on $|\Pi(G)|$ is another avenue of research. There is a class of planar graphs for which $|\Pi(G)| = \Omega(1.442^n)$ [9]. These graphs are biconnected but not triconnected, so it would be interesting to ask if an upper bound smaller than this bound holds for triconnected planar graphs.

References

1. Benjamin Bergougnoux, Mamadou Moustapha Kanté, and Kunihiro Wasa. Disjunctive minimal separators enumeration. In *International Workshop on Enumeration Problems and Applications*, 2019. URL: `https://benjaminbergougnoux.github.io/pdf/bkw19.pdf`.
2. Anne Berry, Jean Paul Bordat, and Olivier Cogis. Generating all the minimal separators of a graph. *Int. J. Found. Comput. Sci.*, 11(3):397–403, 2000. `doi:10.1142/S0129054100000211`.
3. Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, December 1996. `doi:10.1137/S0097539793251219`.
4. Hans L Bodlaender, Leizhen Cai, Jianer Chen, Michael R Fellows, Jan Arne Telle, and Dániel Marx. Open problems in parameterized and exact computation—IWPEC 2006. *Department of Information and Computing Sciences, Utrecht University*, 2006.
5. Hans L. Bodlaender and Arie M. C. A. Koster. Safe separators for treewidth. *Discret. Math.*, 306(3):337–350, 2006. `doi:10.1016/J.DISC.2005.12.017`.
6. Vincent Bouchitté, Frédéric Mazoit, and Ioan Todinca. Chordal embeddings of planar graphs. *Discret. Math.*, 273(1-3):85–102, 2003. `doi:10.1016/S0012-365X(03)00230-9`.
7. Vincent Bouchitté and Ioan Todinca. Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM J. Comput.*, 31(1):212–232, 2001. `doi:10.1137/S0097539799359683`.
8. Vincent Bouchitté and Ioan Todinca. Listing all potential maximal cliques of a graph. *Theor. Comput. Sci.*, 276(1-2):17–32, 2002. `doi:10.1016/S0304-3975(01)00007-X`.
9. Fedor V. Fomin, Dieter Kratsch, Ioan Todinca, and Yngve Villanger. Exact algorithms for treewidth and minimum fill-in. *SIAM J. Comput.*, 38(3):1058–1079, 2008. `doi:10.1137/050643350`.
10. Fedor V. Fomin, Ioan Todinca, and Yngve Villanger. Exact algorithm for the maximum induced planar subgraph problem. In Camil Demetrescu and Magnús M. Halldórsson, editors, *Proceedings of the 19th Annual European Symposium, ESA 2011*, volume 6942 of *Lecture Notes in Computer Science*, pages 287–298. Springer, 2011. `doi:10.1007/978-3-642-23719-5_25`.

11 Fedor V. Fomin, Ioan Todinca, and Yngve Villanger. Large induced subgraphs via triangulations and CMSO. *SIAM J. Comput.*, 44(1):54–87, 2015. doi:10.1137/140964801.

12 Fedor V. Fomin and Yngve Villanger. Finding induced subgraphs via minimal triangulations. In Jean-Yves Marion and Thomas Schwentick, editors, *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4-6, 2010, Nancy, France*, volume 5 of *LIPIcs*, pages 383–394. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPICS.STACS.2010.2470.

13 Fedor V. Fomin and Yngve Villanger. Treewidth computation and extremal combinatorics. *Comb.*, 32(3):289–308, 2012. doi:10.1007/S00493-012-2536-Z.

14 Pinar Heggernes. Minimal triangulations of graphs: A survey. *Discrete Mathematics*, 306(3): 297–317, 2006. doi:10.1016/J.DISC.2005.12.003.

15 John E. Hopcroft and Robert Endre Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974. doi:10.1145/321850.321852.

16 Torsten Inkmann. *Tree-based decompositions of graphs on surfaces and applications to the Traveling Salesman Problem*. PhD thesis, Georgia Institute of Technology, 2008. URL: http://hdl.handle.net/1853/22583.

17 Frank Kammer and Torsten Tholey. Approximate tree decompositions of planar graphs in linear time. *Theoretical Computer Science*, 645:60–90, 2016. doi:10.1016/j.tcs.2016.06.040.

18 Tuukka Korhonen and Daniel Lokshtanov. An improved parameterized algorithm for treewidth. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*, pages 528–541. ACM, 2023. doi:10.1145/3564246.3585245.

19 Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Comb.*, 14(2):217–241, 1994. doi:10.1007/BF01215352.

20 Hisao Tamaki. Positive-instance driven dynamic programming for treewidth. *J. Comb. Optim.*, 37(4):1283–1311, 2019. doi:10.1007/S10878-018-0353-Z.

21 Takeaki Uno and Hiroko Satoh. An efficient algorithm for enumerating chordless cycles and chordless paths. In Saso Dzeroski, Pance Panov, Dragi Kocev, and Ljupco Todorovski, editors, *Proceedings of the 17th International Conference of Discovery Science, DS 2014*, volume 8777 of *Lecture Notes in Computer Science*, pages 313–324. Springer, 2014. doi:10.1007/978-3-319-11812-3_27.

22 Hassler Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55(1):245–254, 1933. doi:10.2307/2371127.