Subexponential-Time and FPT Algorithms for Embedded Flat Clustered Planarity

Giordano Da Lozzo[1], David Eppstein[2], Michael T. Goodrich[2], and Siddharth Gupta[2]

[1] Department of Computer Science, Roma Tre University, Rome, IT dalozzo@dia.uniroma3.it
[2] Department of Computer Science, University of California, Irvine, USA
{eppstein,goodrich,guptasid}@uci.edu

Abstract. The C-Planarity problem asks for a drawing of a *clustered graph*, i.e., a graph whose vertices belong to properly nested clusters, in which each cluster is represented by a simple closed region with no edge-edge crossings, no region-region crossings, and no unnecessary edge-region crossings. We study C-Planarity for *embedded flat clustered graphs*, graphs with a fixed combinatorial embedding whose clusters partition the vertex set. Our main result is a subexponential-time algorithm to test C-Planarity for these graphs when their face size is bounded. Furthermore, we consider a variation of the notion of *embedded tree decomposition* in which, for each face, including the outer face, there is a bag that contains every vertex of the face. We show that C-Planarity is fixed-parameter tractable with the embedded-width of the underlying graph and the number of disconnected clusters as parameters.

1 Introduction

A *clustered graph* (or *c-graph*) is a pair $\mathcal{C}(G, \mathcal{T})$ with *underlying graph* G and *inclusion tree* $\mathcal{T}$, i.e., a rooted tree whose leaves are the vertices of G. Each internal node μ of $\mathcal{T}$ represents a cluster of vertices of G (its leaf descendants) which induces a subgraph $G(\mu)$. A *c-planar drawing* of $\mathcal{C}(G, \mathcal{T})$ (Fig. 1) consists of a drawing of G and of a representation of each cluster μ as a *simple closed region* $R(\mu)$, i.e., a region homeomorphic to a closed disc, such that: (1) Each region $R(\mu)$ contains the drawing of $G(\mu)$. (2) For every two clusters $\mu, \nu \in \mathcal{T}$, $R(\nu) \subseteq R(\mu)$ if and only if ν is a descendant of μ in $\mathcal{T}$. (3) No two edges cross. (4) No edge crosses any region boundary more than once. (5) No two region boundaries intersect.

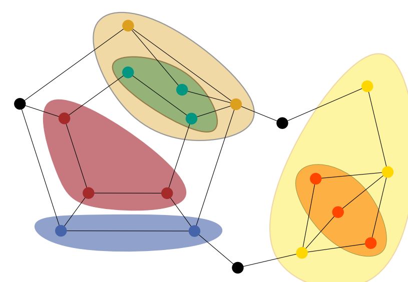


Fig. 1: A c-planar drawing

An interesting and challenging line of research in graph drawing concerns the computational complexity of the C-Planarity problem, which asks to test the existence of a c-planar drawing of a c-graph. This problem is notoriously difficult, particularly when (as in Fig. 1) clusters may be disconnected, faces may have unbounded size, and the cluster hierarchy may have multiple nested levels. No known subexponential-time algorithm solves the (general) C-Planarity problem, and it is unknown whether it is NP-complete, although the related problem of splitting as few clusters as possible to make a c-graph c-planar was proved NP-hard [5]. Thus, there is considerable interest in subexponential-time, slice-wise polynomial, and fixed-parameter tractable algorithms, besides polynomial-time algorithms for special cases of C-Planarity.

C-Planarity was introduced by Feng, Cohen, and Eades [24], who solved it in quadratic time for the *c-connected* case when every cluster induces a connected subgraph. Similar results were given by Lengauer [32] using different terminology. Dahlhaus [21] claimed a linear-time algorithm for c-connected C-Planarity (with some details later provided by Cortese *et al.* [18]). Goodrich *et al.* [27] gave a cubic-time algorithm for disconnected clusters that satisfy an "extroverted" property, and Gutwenger *et al.* [28] provided a polynomial-time algorithm for "almost" c-connected inputs. Cornelsen and Wagner showed polynomiality for completely connected c-graphs, i.e., c-graphs for which not only every cluster but also the complement of each cluster is connected [17]. FPT algorithms have also been investigated [10,15]. For additional special cases, see, e.g., [2,3,4,7,14,23].

A c-graph is *flat* when no non-trivial cluster is a subset of another, so $\mathcal{T}$ has only three levels: the root, the clusters, and the leaves. Flat C-Planarity can be solved in polynomial time for embedded c-graphs with at most 5 vertices per face [22,26] or at most two vertices of each cluster per face [13], for embedded c-graphs in which each cluster induces a subgraph with at most two connected components [30], and for c-graphs with two clusters [9,26,29] or three clusters [1]. At the other end of the size spectrum, Jelínková *et al.* [31] provide efficient algorithms for 3-connected flat c-graphs when each cluster has at most 3 vertices. Fulek [25] speculates that C-Planarity could be solvable in subexponential time for more general embedded flat c-graphs.

New Results. In this paper, we provide subexponential-time and fixed-parameter tractable algorithms for broad classes of c-graphs. We show the following results:

- C-Planarity can be solved in subexponential time for embedded flat c-graphs with bounded face size (Section 3).
- C-Planarity is fixed-parameter tractable for embedded flat c-graphs with embedded-width and number of disconnected clusters as parameters (Section 4).

Our first result uses divide-and-conquer with a large but subexponential branching factor. It exploits cycle separators in planar graphs and a concise representation of the connectivity of each cluster in a c-planar drawing. This method also leads to an XP algorithm for *generalized h-simply nested graphs*, which extend simply-nested graphs with bounded face size (Section 3.1). Recall that, XP (short for *slice-wise polynomial*) is the class of parameterized problems with input size n and parameter k than can be solved in $O(n^{f(k)})$ time, where f is a computable function.

We obtain our second result by expressing c-planarity in extended monadic second-order logic for embedded flat c-graphs and applying Courcelle's Theorem. The graphs to which this result applies, with bounded treewidth and bounded face size, include the *nested triangles graphs*, a standard family of examples that are hard for many graph drawing tasks, the *dual graphs of bounded-treewidth bounded-degree plane graphs* [12], and the *buckytubes*, graphs formed from a planar hexagonal lattice wrapped to form a cylinder of bounded diameter.

2 Definitions and Preliminaries

The graphs considered in this paper are finite, simple, and connected. A graph is *planar* if it admits a drawing in the plane without edge crossings. A *combinatorial embedding* is an equivalence class of planar drawings, where two drawings of a graph are *equivalent* if they determine the same *rotation* at each vertex, i.e, the same circular orderings for the edges around each vertex. A planar drawing partitions the plane into topologically connected regions, called *faces*. The bounded faces are the *inner faces*, while the unbounded face is the *outer face*. A combinatorial embedding together with a choice for the outer face defines a *planar embedding*. An *embedded graph* (*plane graph*) is a planar graph with a fixed combinatorial embedding (fixed planar embedding). The *length* of a face f is the number of occurrences of edges encountered in a traversal of f. The *maximum face size* of an embedded graph is the length of its largest face.

A graph is *connected* if it contains a path between any two vertices. A *cut-vertex* is a vertex whose removal disconnects the graph. A *separation pair* is a pair of vertices whose removal disconnects the graph. A connected graph is 2-*connected* if it contains at least 3 vertices and does not have a cut-vertex, and a 2-connected graph is 3-*connected* if it contains at least 4 vertices and does not have a separation pair. The *blocks* of a graph are its maximal 2-connected subgraphs. Any (subdivision of a) 3-connected planar graph admits a unique combinatorial embedding (up to a flip) [34].

Tree-width and Embedded-width. A *tree decomposition* of a graph G is a tree T whose nodes, called *bags*, are labeled by subsets of vertices of G. For each vertex v the bags containing v must form a nonempty contiguous subtree of T, and for each edge uv at least one bag must contain both u and v. The *width* of the decomposition is one less than the maximum cardinality of any bag, and the *treewidth* $\mathrm{tw}(G)$ of G is the minimum width of any of its tree decompositions.

Recently, Borradaile *et al.* [11] developed a variant of treewidth, specialized for plane graphs, called embedded-width. According to their definitions, a tree decomposition *respects* an embedding of a plane graph G if, for every inner face f of G, at least one bag contains all the vertices of f. They define the *embedded-width* $\mathrm{emw}(G)$ of G to be the minimum width of a tree decomposition that respects the embedding of G. We will use the following result [11].

Theorem 1 ([11], Theorem 2). *If G is a plane graph where every inner face has length at most ℓ, then* $\mathrm{emw}(G) \leq (\mathrm{tw}(G)+2)\cdot \ell - 1$.

Borradaile *et al.* do not require the vertices of the outer face to be contained in a same bag. In our applications, we modify this concept so that the tree decomposition also includes a bag containing the outer face, and we denote the minimum width of such a tree decomposition as $\overline{\mathrm{emw}}(G)$. We have the following.

Lemma 1. *If G is a plane graph whose maximum face size (including the size of the outer face) is ℓ, then* $\overline{\mathrm{emw}}(G) = O(\ell \cdot \mathrm{tw}(G))$.

Proof. To prove the statement, we can proceed as follows.

We augment G to a graph G', by embedding G in the interior of a triangle Δ and by identifying one of the vertices of the outer face of G with a vertex of Δ. Clearly, $tw(G') = \max(\mathrm{tw}(G), 2)$ and G' has maximum face size

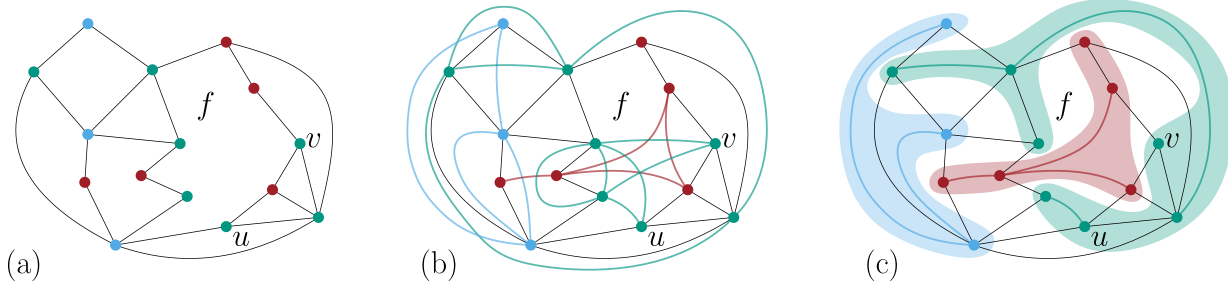


Fig. 2: (a) An embedded flat c-graph $\mathcal{C}(G,\mathcal{T})$. (b) A super c-graph of $\mathcal{C}$ containing all the candidate saturating edges of $\mathcal{C}$ (thick and colored curves); since vertices u and v belong to different components of $X_\mu(f)$ but to the same connected component of $G(\mu)$, edge (u,v) is not a candidate saturating edge. (c) A super c-graph of $\mathcal{C}$ satisfying Condition (iii) of Theorem 2; regions enclosing vertices of each cluster are shaded.

$\ell' = O(\ell)$. Also, we have that $\overline{\mathrm{emw}}(G) \leq \mathrm{emw}(G')$, since $G \subseteq G'$ and since all the vertices incident to the same face in G are also incident to the same face in G'. Thus, the statement follows from the fact that, by Theorem 1, $\mathrm{emw}(G') \leq (\mathrm{tw}(G')+2)\cdot \ell' - 1$. □

Clustered Planarity. Recall that, in a c-graph $\mathcal{C}(G,\mathcal{T})$, each internal node μ of $\mathcal{T}$ corresponds to the set $V(\mu)$ of vertices of G at leaves of the subtree of $\mathcal{T}$ rooted at μ. Set $V(\mu)$ induces the subgraph $G(\mu)$ of G. We call the internal nodes other than the root *clusters*. A cluster μ is *connected* if $G(\mu)$ is connected and *disconnected* otherwise. A c-graph $\mathcal{C}(G,\mathcal{T})$ is *c-connected* if every cluster is connected.

A c-graph is *c-planar* if it admits a c-planar drawing. Two c-graphs $\mathcal{C}(G,\mathcal{T})$ and $\mathcal{C}'(G',\mathcal{T}')$ are *equivalent* if both are c-planar or neither is. If the root of $\mathcal{T}$ has leaf children, enclosing each leaf v in a new singleton cluster produces an equivalent c-graph. Therefore, we can safely assume that each vertex belongs to a cluster. A c-graph is *flat* if each leaf-to-root path in $\mathcal{T}$ has exactly three nodes. The clusters of a flat c-graph form a partition of the vertex set.

An *embedded c-graph* $\mathcal{C}(G,\mathcal{T})$ is a c-graph whose underlying graph has a fixed combinatorial embedding. It is *c-planar* if it admits a c-planar drawing that preserves the embedding of G. In what follows, we only deal with embedded flat c-graphs. Therefore, we will refer to such graphs simply as c-graphs.

We define the *candidate saturating edges* of a c-graph $\mathcal{C}(G,\mathcal{T})$ as follows. For each face f of G, let $G(f)$ be the closed walk composed of the vertices and edges of f. For each cluster $\mu \in \mathcal{T}$, consider the set $\mathcal{X}_\mu(f)$ of connected components of $G(f)$ induced by the vertices of μ and, for each component $\xi \in \mathcal{X}_\mu(f)$, assign a vertex of f belonging to ξ as a reference vertex of ξ. We add an edge inside f between the reference vertices of any two components in $\mathcal{X}_\mu(f)$ if and only if such vertices belong to different connected components of $G(\mu)$; see Figs. 2a and 2b. A c-graph obtained from $\mathcal{C}(G,\mathcal{T})$ by adding to $\mathcal{C}$ a subset E^+ of its candidate saturating edges is a *super c-graph* of $\mathcal{C}$.

Di Battista and Frati [22] gave the following characterization.

Theorem 2 ([22], Theorem 1). *A c-graph $\mathcal{C}(G,\mathcal{T})$ is c-planar if and only if:*

(i) G is planar;
(ii) there exists a face f in G such that when f is chosen as the outer face for G no cycle composed of vertices of the same cluster encloses a vertex of a different cluster in its interior; and
(iii) there exists a super c-graph $\mathcal{C}'(G',\mathcal{T})$ of $\mathcal{C}$ such that G' is planar and $\mathcal{C}'$ is c-connected (see Fig. 2c).

Conditions (i) and (ii) of Theorem 2 can be easily verified in linear time. Therefore, we can assume that any c-graph satisfies these conditions. Following [22] we thus view the problem of testing c-planarity as one of testing Condition (iii).

A *cluster-separator* in a c-graph $\mathcal{C}(G,\mathcal{T})$ is a cycle ρ in G for which some cluster $\mu \in \mathcal{T}$ has vertices both in the interior and in the exterior of ρ but with $V(\mu)\cap V(\rho) = \emptyset$. Condition (iii) immediately yields the following observation.

Observation 1. *A c-graph that has a cluster-separator is not c-planar.*

(a) (b)

Fig. 3: Transformations for the proof of Lemma 2.

In the next sections, it will be useful to only consider c-graphs which are at least 2-connected (Section 3) and 3-connected (Section 4). The next lemma, conveniently stated in a stronger form[3], shows that this is not a loss of generality.

Lemma 2. *Let $\mathcal{C}(G,\mathcal{T})$ be an n-vertex c-graph with maximum face size ℓ. There exists an $O(n)$-time algorithm that constructs an equivalent c-graph $\mathcal{C}^*(G^*,\mathcal{T}^*)$ with $|V(G^*)| = O(n)$ such that: 1. G^* is 3-connected, 2. the maximum face size κ of G^* is $O(\ell)$, and 3. the c-graph $\mathcal{C}^\diamond(G^\diamond,\mathcal{T}^\diamond)$ obtained by augmenting $\mathcal{C}^*(G^*,\mathcal{T}^*)$ with all its candidate saturating edges is such that $\mathrm{tw}(G^\diamond) = O(\overline{\mathrm{emw}}(G))$.*

Proof. To prove the statement, we can proceed as follows.

First, we transform c-graph $\mathcal{C}(G,\mathcal{T})$ into an equivalent c-graph $\mathcal{C}^{'}(G',\mathcal{T}^{'})$, by applying the transformation in Fig. 3a to every edge, such that $|V(G')| = O(|V(G)|)$, every vertex of G' has degree at least 3, the maximum face size ℓ' of G' is $O(\ell)$, $tw(G') = \mathrm{tw}(G)$, and each vertex u of G' is incident to at least three edges in each block u belongs to.

Second, we transform c-graph $\mathcal{C}^{'}(G',\mathcal{T}^{'})$ into an equivalent c-graph $\mathcal{C}^*(G^*,\mathcal{T}^*)$, by applying the transformation in Fig. 3b to every vertex, such that $|V(G^*)| = O(|V(G')|)$, G^* is 3-connected, the maximum face size ℓ^* of G^* is $O(\ell')$, and the c-graph $\mathcal{C}^\diamond(G^\diamond,\mathcal{T}^\diamond)$ obtained by augmenting $\mathcal{C}^*(G^*,\mathcal{T}^*)$ with all its candidate saturating edges is such that $\mathrm{tw}(G^\diamond) = O(\ell \cdot \mathrm{tw}(G))$, which implies that $\mathrm{tw}(G^\diamond) = \overline{\mathrm{emw}}(G)$, since $\overline{\mathrm{emw}}(G) = O(\ell \cdot \mathrm{tw}(G))$ by Lemma 1.

We now describe each of the transformations in detail.

First, initialize $\mathcal{C}^{'} = \mathcal{C}$. For every vertex c of G, let (c,x) be any edge incident to c. Add to G' vertices w_1 and w_2 and embed paths (c,w_1,x) and (c,w_2,x) in the interior of each of the two faces of G' edge (c,x) is incident to; also, subdivide edge (c,x) with a vertex w, add edges (w_1,w) and (w_2,w), and assign vertices w_1, w_2, and w to the same cluster of $\mathcal{T}^{'}$ $(\mathcal{T})$ vertex c belongs to. Refer to Fig. 3a. By construction, all the newly added vertices have degree at least 3. In particular, observe that each cut-vertex of G' is incident to at least three edges in each of the blocks such a cut-vertex belongs to. It is easy to see that $\mathcal{C}^{'}$ and $\mathcal{C}$ are equivalent. Also, the maximum face size ℓ' of G' is $O(\ell)$. Further, $\mathrm{tw}(G') = \max(\mathrm{tw}(G), 3)$, as the transformation replaces edges with subgraphs of treewidth 3.

Second, initialize $\mathcal{C}^* = \mathcal{C}^{'}$. For every vertex c of G', we subdivide each edge (c,x_i) incident to c with a dummy vertex v_i. Denote such a graph by G^+. Also, add an edge between any two vertices v_i and v_j such that edges (c,x_i) and (c,x_j) are consecutive around c in the unique face shared by v_i and v_j in G^+. Finally, assign each vertex v_i to the same cluster of $\mathcal{T}^*$ $(\mathcal{T}^{'})$ vertex c belongs to. Refer to Fig. 3b. The equivalence between $\mathcal{C}^*(G^*,\mathcal{T}^*)$ and $\mathcal{C}^{'}(G',\mathcal{T}^{'})$ is again straightforward. Clearly, $|V(G^*)| = O(|V(G')|)$ and $\ell^* = O(\ell')$. Also, by the observation that the cut-vertices of G' are incident to at least three edges in each of the blocks such cut-vertices belong to, the applied transformation fixes the rotation at all the vertices of G^*. Since each vertex of G^* has minimum degree 3 and G^* has a fixed combinatorial embedding (up to a flip), by the result of Whitney [34], we have that G^* is 3-connected. Furthermore, $\overline{\mathrm{emw}}(G^*) = O(\overline{\mathrm{emw}}(G'))$, since G^* is obtained by subdividing each edge of G' twice, thus obtaining a graph G^+ with the same tree-width as G' and maximum face-size in $O(\ell')$, and by adding edges between some of the vertices incident to the faces of G^+. Since $G^\diamond$ is the graph obtained by adding all the candidate saturating-edges of G^* (recall that such edges only connect vertices in the same face of G^*), we have that $\mathrm{tw}(G^\diamond) = O(\overline{\mathrm{emw}}(G^*)) = O(\overline{\mathrm{emw}}(G'))$. Since, by Lemma 1, $\overline{\mathrm{emw}}(G') = O(\ell' \cdot \mathrm{tw}(G'))$ and since $\ell' = O(\ell)$ and $\mathrm{tw}(G') = O(\mathrm{tw}(G))$, we have that $\mathrm{tw}(G^\diamond) = O(\ell \cdot \mathrm{tw}(G))$. This concludes the proof of the lemma. ☐

[3] In Section 4, we exploit all the properties of the lemma. In Section 3, we only exploit the existence of an equivalent 2-connected c-graph with maximum face size $\kappa = O(\ell)$.

3 A Subexponential-Time Algorithm for C-Planarity

In this section, we describe a divide-and-conquer algorithm for testing the c-planarity of 2-connected c-graphs exploiting cycle separators in planar graphs.

The "conquer" part of our divide-and-conquer uses the following operation on pairs of c-graphs. Let G_1 and G_2 be plane graphs on overlapping vertex sets such that the outer face of G_1 and an inner face of G_2 are bounded by the same cycle ρ. *Merging* G_1 and G_2 constructs a new plane graph G from $G_1 \cup G_2$ as follows. We remove multi-edges (belonging to cycle ρ) and assign each vertex v a rotation whose restriction to the edges of G_2 (of G_1) is the same as the rotation at v in G_2 (in G_1). This is possible as cycle ρ bounds the outer face of G_1 and an inner face of G_2. We say that G is a *merge* of G_1 and G_2. Now consider two c-graphs $\mathcal{C}_1(G_1, \mathcal{T}_1)$ and $\mathcal{C}_2(G_2, \mathcal{T}_2)$ such that (i) $G_1 \cap G_2 = \rho$ is a cycle, (ii) for each vertex $v \in V(\rho)$, vertex v belongs to the same cluster μ both in $\mathcal{T}_1$ and in $\mathcal{T}_2$, and (iii) cycle ρ bounds the outer face of G_1 and an inner face of G_2 (when a choice for their outer faces that satisfies Condition (ii) of Theorem 2 has been made). *Merging* $\mathcal{C}_1$ and $\mathcal{C}_2$ is the operation that constructs a c-graph $\mathcal{C}(G, \mathcal{T})$ as follows. Graph G is obtained by merging G_1 and G_2. Tree $\mathcal{T}$ is obtained as follows. Initialize $\mathcal{T}$ to $\mathcal{T}_1$. First, for each cluster $\mu \in \mathcal{T}_2 \cap \mathcal{T}_1$, we add the leaves of μ in $\mathcal{T}_2$ as children of cluster μ in $\mathcal{T}$, removing duplicate leaves. Second, for each cluster $\mu \in \mathcal{T}_2 \setminus \mathcal{T}_1$, we add the subtree of $\mathcal{T}_2$ rooted at μ as a child of the root of $\mathcal{T}$. We say that $\mathcal{C}(G, \mathcal{T})$ is a *merge* of $\mathcal{C}_1(G_1, \mathcal{T}_1)$ and $\mathcal{C}_2(G_2, \mathcal{T}_2)$.

In the "divide" part of the divide-and-conquer, we replace subgraphs of the input by smaller planar components called *cycle-stars* that preserve their c-planarity properties. Let G be a connected plane graph that contains a face whose boundary is a cycle ρ. We say that G is a *cycle-star* if removing all the edges of ρ makes G a forest of stars; refer to Fig. 4c. Also, we say that cycle ρ is *universal* for G and we say that a vertex of G is a *star vertex* of G if it does not belong to ρ. Clearly, the size of G is $O(|\rho|)$.

For a c-planar c-graph $\mathcal{C}(G, \mathcal{T})$ and a cycle separator ρ, we denote by $\mathcal{C}^+_\rho(G^+, \mathcal{T}^+)$ (by $\mathcal{C}^-_\rho(G^-, \mathcal{T}^-)$) the c-graph obtained from $\mathcal{C}$ by removing all the vertices and the edges of G that lie in the interior of ρ (in the exterior of ρ). Consider a super c-graph $\mathcal{C}'(G', \mathcal{T})$ of $\mathcal{C}$ satisfying Condition (iii) of Theorem 2, which exists since $\mathcal{C}$ is c-planar. We now give a procedure, which will be useful throughout the paper, to construct two special c-planar c-graphs $\mathcal{C}^-(S^-, \mathcal{K}^-)$ and $\mathcal{C}^+(S^+, \mathcal{K}^+)$ associated with $\mathcal{C}'$ whose properties are described in the following lemma.

Lemma 3. *C-graphs $\mathcal{C}^-(S^-, \mathcal{K}^-)$ and $\mathcal{C}^+(S^+, \mathcal{K}^+)$ are such that:*

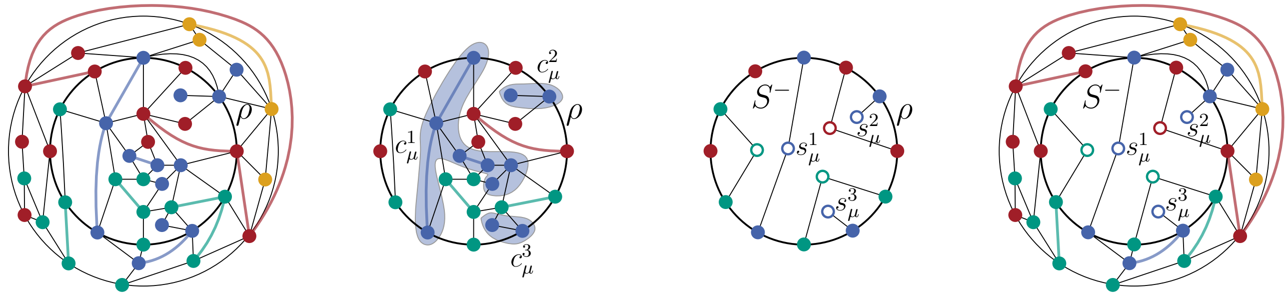
1. *graph S^- (S^+) is a cycle-star whose universal cycle is ρ,*
2. *cycle ρ bounds the outer face of S^- (an inner face of S^+),*
3. *each star vertex of S^- (S^+) and all its neighbours belong to the same cluster in $\mathcal{K}^-$ ($\mathcal{K}^+$), and*
4. *the c-graph $\mathcal{C}_{out}$ ($\mathcal{C}_{in}$) obtained by merging $\mathcal{C}^-(S^-, \mathcal{K}^-)$ and $\mathcal{C}^+_\rho(G^+, \mathcal{T}^+)$ (by merging $\mathcal{C}^+(S^+, \mathcal{K}^+)$ and $\mathcal{C}^-_\rho(G^-, \mathcal{T}^-)$) is c-planar.*

We describe how to construct $\mathcal{C}^-(S^-, \mathcal{K}^-)$ from $\mathcal{C}'$; refer to Fig. 4. The construction of $\mathcal{C}^+(S^+, \mathcal{K}^+)$ is symmetric.

First, for each cluster μ such that $V(\mu) \cap V(\rho) = \emptyset$, we remove all the vertices in $V(\mu)$ lying in the interior of ρ together with their incident edges. By Observation 1, the resulting c-graph is still c-planar and c-connected. Also, we remove edges in the interior of ρ whose endpoints belong to different clusters. Clearly, this simplification preserve c-connectedness. We still denote the resulting c-graph as $\mathcal{C}'$.

Second, consider the c-graph H consisting of the vertices and of the edges of $\mathcal{C}'$ lying in the interior and along the boundary of ρ. For each cluster μ and for each connected component c^i_μ of μ in H, we replace all the vertices and edges of c^i_μ lying in the interior of ρ in $\mathcal{C}'$ with a single vertex s^i_μ, assigning it to the same cluster μ and making it adjacent to all the vertices in $V(c^i_\mu) \cap V(\rho)$. Let $\mathcal{C}^*$ be the resulting c-graph. It is easy to see that such a transformation preserves c-connectedness and planarity, therefore $\mathcal{C}^*$ is a c-connected c-planar c-graph. By construction, each vertex $v \in V(\rho)$ is adjacent to a single vertex s^i_μ, where μ is the cluster vertex v belongs to; thus, the vertices and the edges in the interior and along the boundary of ρ in $\mathcal{C}^*$ form c-graph $\mathcal{C}^-(S^-, \mathcal{K}^-)$ whose underlying graph S^- is a cycle-star satisfying Properties (1), (2) and (3) of Lemma 3. Further, since the subgraph of $\mathcal{C}^*$ consisting of the vertices and of the edges lying in the exterior and along the boundary of ρ coincides with $\mathcal{C}^+_\rho(G^+, \mathcal{T}^+)$, we have that $\mathcal{C}^*$ is a c-planar c-connected super c-graph of $\mathcal{C}_{out}$. Thus, by Condition (iii) of Theorem 2, Property (4) of Lemma 3 is also satisfied.

Let $\mathcal{C}^-_\Delta(R^-, \mathcal{J}^-)$ ($\mathcal{C}^+_\Delta(R^+, \mathcal{J}^+)$) be a c-graph obtained by augmenting the c-graph $\mathcal{C}^-(S^-, \mathcal{K}^-)$ ($\mathcal{C}^+(S^+, \mathcal{K}^+)$) of Lemma 3 by introducing new vertices, each belonging to a distinct cluster, and by adding edges only between the vertices in $V(S^-)$ ($V(S^+)$) and the newly introduced vertices in such a way that cycle ρ bounds a face of R^- (R^+) and all the other faces of R^- (R^+) are triangles. From the construction of Lemma 3, we also have the following useful technical remark.



(a) $\mathcal{C}'$ **(b)** H **(c)** S^- **(d)** $\mathcal{C}^*$

Fig. 4: (a) Super c-graph $\mathcal{C}'$ of $\mathcal{C}$. (b) Each component of the blue cluster μ in H lies inside a simple closed region. (c) Cycle-star S^- corresponding to H. (d) The c-connected c-planar c-graph $\mathcal{C}^*$ obtained by *replacing* H with S^- in $\mathcal{C}'$.

Remark 1. *The c-graph obtained by merging $\mathcal{C}^-_\Delta(R^-, \mathcal{J}^-)$ and $\mathcal{C}^+_\rho(G^+, \mathcal{T}^+)$ (by merging $\mathcal{C}^+_\Delta(R^+, \mathcal{J}^+)$ and $\mathcal{C}^-_\rho(G^-, \mathcal{T}^-)$) is c-planar.*

We now describe a divide-and-conquer algorithm based on Lemma 3, called TestCP, that tests the c-planarity of a 2-connected c-graph $\mathcal{C}(G, \mathcal{T})$ and returns a super c-graph $\mathcal{C}^*(G^*, \mathcal{T})$ of $\mathcal{C}$ satisfying Condition (iii) of Theorem 2, if $\mathcal{C}$ is c-planar. See Fig. 5 for illustrations of the c-graphs constructed during the execution of the algorithm.

We first give an intuition on the role of cycle-stars in Algorithm TestCP.

Let $\mathcal{C}(G, \mathcal{T})$ be a c-planar c-graph and let ρ be a cycle separator of G. By Lemma 3, for each c-connected c-planar super c-graph $\mathcal{C}'$ of $\mathcal{C}$, we can injectively map the super c-graph I^- of $\mathcal{C}^-_\rho(G^-, \mathcal{T}^-)$, composed of the vertices of G^- and of the edges in the interior and along the boundary of ρ in $\mathcal{C}'$, with a cycle-star S^- whose universal cycle is ρ. This is due to the fact that there exists a *one-to-one correspondence* between the connected components of I^- induced by the vertices of each cluster in $\mathcal{T}^-$ and the star vertices of S^-. Similar considerations hold for the super c-graph I^+ of $\mathcal{C}^+_\rho(G^+, \mathcal{T}^+)$. Although the c-planarity of $\mathcal{C}^+_\rho$ and $\mathcal{C}^-_\rho$ is necessary for the c-planarity of $\mathcal{C}$, it is not a sufficient condition, as the connectivity of clusters inside ρ in I^- (*internal cluster-connectivity*) and the connectivity of clusters outside ρ in I^+ (*external cluster-connectivity*) must also together determine the c-connectedness of $\mathcal{C}'$. The role of cycle-stars S^- and S^+ in the algorithm presented in this section is exactly that of concisely *representing* the internal cluster-connectivity of I^- and the external cluster-connectivity of I^+, respectively, to devise a divide-and-conquer approach to test the c-planarity of $\mathcal{C}$.

Outline of the algorithm. We overview the main steps of our algorithm below.

- If $n = O(\ell)$, we test c-planarity directly, as a base case for the divide-and-conquer recursion. Otherwise, we construct a cycle-separator ρ of G and test whether ρ is a cluster-separator. If so, $\mathcal{C}$ cannot be c-planar (Observation 1), and we halt the search.
- We generate all cycle-stars S_i^- with universal cycle ρ. A cycle-star S_i^- represents a potential connection pattern of clusters inside ρ. For each cycle-star S_i^- we apply Procedure OuterCheck to test whether this pattern could be augmented by additional connections outside ρ to complete the desired cluster-connectivity. That is, we test whether $\mathcal{C}^+_\rho$ admits a c-connected c-planar super c-graph whose internal cluster-connectivity is represented by S_i^-. To test this, we replace the subgraph G^- of G in $\mathcal{C}$ with an internally-triangulated supergraph R_i^- of S_i^- to obtain a c-graph $\mathcal{C}^+$ and recursively test $\mathcal{C}^+$ for c-planarity. It is important to observe that, the triangulation step prevents $\mathcal{C}^+$ from having saturating edges inside ρ, thus enforcing exactly the same internal-cluster connectivity represented by S_i^- (Remark 1). If $\mathcal{C}^+$ is c-planar, the procedure returns a c-connected c-planar super c-graph $\mathcal{C}^+_{con}$ of $\mathcal{C}^+$. If no cycle-star passes the test, $\mathcal{C}$ is not c-planar by Lemma 3. We call all the cycle-stars that passed this test *admissible*.
- We then apply Procedure InnerCheck to verify whether the internal-cluster connectivity represented by some admissible cycle-star S_i^- can *actually* be realized by a c-connected c-planar super c-graph of $\mathcal{C}$. For each admissible cycle-star S_i^-, the procedure applies the construction of Lemma 3 to obtain a cycle-star S_i^+ representing the external cluster-connectivity of $\mathcal{C}^+_{con}$. Then, it tests whether $\mathcal{C}^-_\rho$ admits a c-connected c-planar super c-graph $\mathcal{C}^-_{con}$ whose external cluster-connectivity is represented by S_i^+. This is done similarly to Procedure OuterCheck, by triangulating the exterior of ρ and recursively testing c-planarity of a smaller graph. If Procedure InnerCheck succeeds for any admissible cycle-star S_i^-, then we can merge the subgraphs of $\mathcal{C}^-_{con}$ and of $\mathcal{C}^+_{con}$ induced by the

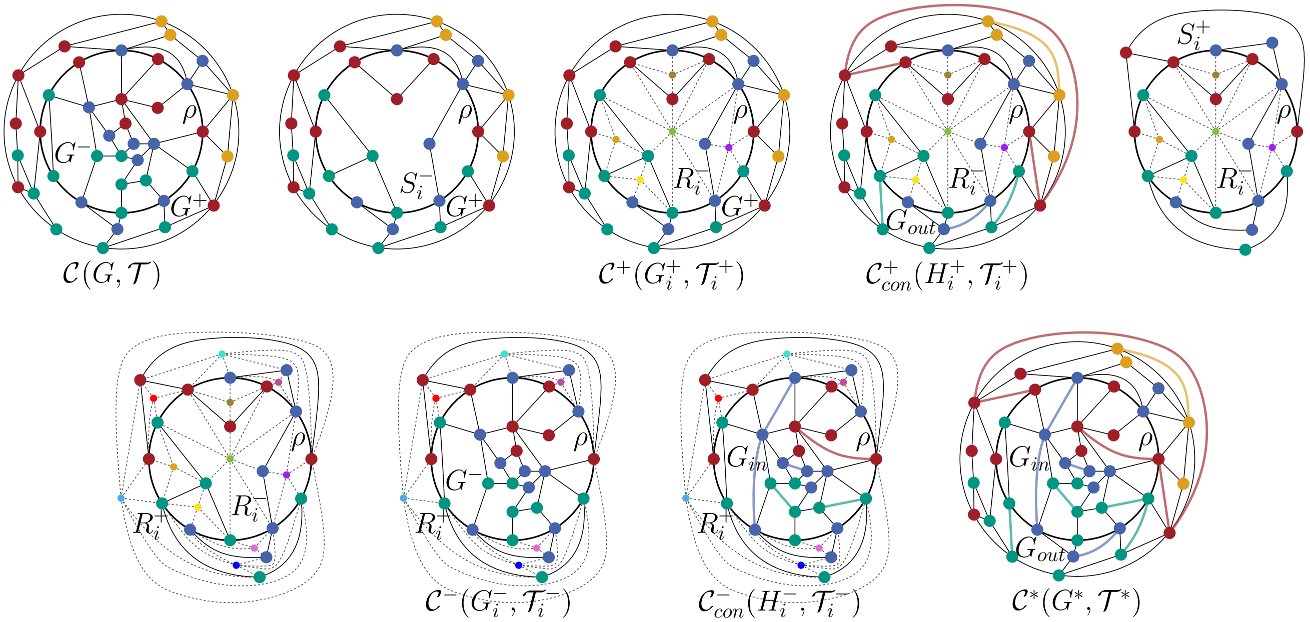


Fig. 5: Illustrations of all of the c-graphs constructed by Algorithm TestCP.

vertices inside and outside ρ, respectively, to obtain a c-connected c-planar super c-graph of $\mathcal{C}$, and we halt the search with a successful outcome. It might be the case that $\mathcal{C}^-_{con}$ has a different internal-cluster connectivity than that represented by S_i^-, but this is not a problem, because the different cluster connectivity (which necessarily corresponds to a different admissible cycle-star) still provides a c-planar drawing of the whole graph.

– If no admissible cycle-star passes Procedure InnerCheck, $\mathcal{C}$ is not c-planar.

It is crucial in this algorithm that ρ be a cycle-separator. Because it is a cycle, no candidate saturating edges can connect vertices in the interior of ρ to vertices in the exterior of ρ, as such vertices do not share any face. That is, the interaction between G^-_ρ and G^+_ρ only happens through vertices of ρ. This allows us to split the instance into smaller instances recursively along ρ and model the interaction via cycle-stars (by Lemma 3 and Remark 1) whose universal cycle is ρ.

The complete listing of Algorithm TestCP is provided in the next page.

Base Case of the algorithm. The base case occurs when $\mathcal{C}^+(G_i^+, \mathcal{T}_i^+)$ and $\mathcal{C}^-(G_i^-, \mathcal{T}_i^-)$ are no longer smaller than $\mathcal{C}(G, \mathcal{T})$.

Observe that, we obtained G_i^+ (G_i^-) by merging G^+ (G^-) and R_i^- (R_i^+) along cycle ρ, which has size $s(n)$. The size of G^+ and G^- is bounded by $\frac{2n}{3} + s(n)$, while the size of R_i^- and R_i^+ is bounded by $3s(n)$. Therefore, since cycle ρ is shared by all the mentioned graphs by construction, we have that the size of G_i^+ and G_i^- is at most $\frac{2n}{3} + 3s(n)$. Thus, with $s(n) \leq 2\sqrt{\ell n}$ [33], we can set the base case of Algorithm TestCP when $n \leq \frac{2n}{3} + 6\sqrt{\ell n}$, that is, $n \leq 324\ell$.

Correctness of the algorithm. We show that, given a 2-connected c-graph $\mathcal{C}(G, \mathcal{T})$, Algorithm TestCP returns `YES`, which happens when both procedures OuterCheck and InnerCheck succeed, if and only if $\mathcal{C}(G, \mathcal{T})$ is c-planar.

($\Rightarrow$) Suppose that OuterCheck and InnerCheck succeed for a cycle-star $S_\omega^- \in \mathcal{S}$ constructed at step 2a. We show that $\mathcal{C}(G, \mathcal{T})$ is c-planar. Consider the c-graph $\mathcal{C}^*(G^*, \mathcal{T})$ constructed at step 3(a)v from $\mathcal{C}^-_{con}(H_\omega^-, \mathcal{T}_\omega^-)$ and $\mathcal{C}^+_{con}(H_\omega^+, \mathcal{T}_\omega^+)$. The proof of this direction follows by the next claim about $\mathcal{C}^*$ and from Theorem 2.

[1] The merging operations are well defined as cycle ρ bounds the outer face of R_i^- and an inner face of G^+, as well as an inner face of R_i^+ and the outer face of G^-.

[2] As $\mathcal{C}^+(G_i^+, \mathcal{T}_i^+)$ and $\mathcal{C}^-(G_i^-, \mathcal{T}_i^-)$ are 2-connected, TestCP can be recursively applied.

ALGORITHM TESTCP(C-GRAPH $\mathcal{C}(G,\mathcal{T})$)

BASE CASE

If $|V(G)| = O(\ell)$, then we can test C-PLANARITY for $\mathcal{C}(G,\mathcal{T})$ in $O(1)$ time when ℓ is a constant, by performing a brute force search to find a subset E' of the candidate saturating edges of $\mathcal{C}$ such that c-graph $\mathcal{C}'(G \cup E', \mathcal{T})$ satisfies Condition (iii) of Theorem 2.

RECURSIVE STEP

1. Select a cycle separator ρ of G. If ρ is a cluster-separator, then **return** `NO`; otherwise, construct c-graphs $\mathcal{C}^+_\rho(G^+,\mathcal{T}^+)$ and $\mathcal{C}^-_\rho(G^-,\mathcal{T}^-)$ as defined in Lemma 3.
2. OUTERCHECK
   (a) Construct the set $\mathcal{S}$ of all cycle-stars such that, for every $S \in \mathcal{S}$, it holds that (i) ρ is the universal cycle of S, (ii) ρ bounds the outer face of S, and (iii) every star vertex of S is incident only to vertices of ρ belonging to the same cluster.
   (b) For each cycle-star $S^-_i \in \mathcal{S}$:
      i. Construct a c-graph $\mathcal{C}^-(S^-_i,\mathcal{K}^-_i)$ as follows. First, initialize $\mathcal{K}^-_i$ to the subtree of $\mathcal{T}$ whose leaves are the vertices of S^-_i. Then, for each star vertex v of S^-_i, assign v to the cluster $\mu \in \mathcal{K}^-_i$ to which all its neighbours belong.
      ii. Augment $\mathcal{C}^-(S^-_i,\mathcal{K}^-_i)$ to an internally triangulated c-graph $\mathcal{C}^-_\Delta(R^-_i,\mathcal{J}^-_i)$ by introducing new vertices, each belonging to a distinct cluster, and by adding edges only between vertices in $V(S^-_i)$ and the newly introduced vertices (that is, no two non-adjacent vertices in S^-_i are adjacent in R^-_i).
      iii. Merge $\mathcal{C}^-_\Delta(R^-_i,\mathcal{J}^-_i)$ and $\mathcal{C}^+_\rho(G^+,\mathcal{T}^+)$ to obtain a c-graph $\mathcal{C}^+(G^+_i,\mathcal{T}^+_i)$[1].
      iv. Run TESTCP($\mathcal{C}^+(G^+_i,\mathcal{T}^+_i)$) to test whether $\mathcal{C}^+(G^+_i,\mathcal{T}^+_i)$ is c-planar[2].
   (c) If no c-graph $\mathcal{C}^+(G^+_i,\mathcal{T}^+_i)$ is c-planar, then **return** `NO`; otherwise, initialize $\mathcal{S}'$ as the set of *admissible* cycle-stars, i.e., the cycle-stars in $\mathcal{S}$ whose corresponding c-graph $\mathcal{C}^+(G^+_i,\mathcal{T}^+_i)$ is c-planar.
3. INNERCHECK
   (a) For each admissible cycle-star $S^-_i \in \mathcal{S}'$:
      i. Let $\mathcal{C}^+_{con}(H^+_i,\mathcal{T}^+_i)$ be the c-planar c-connected super c-graph of $\mathcal{C}^+$ returned by TESTCP($\mathcal{C}^+(G^+_i,\mathcal{T}^+_i)$) (step 2(b)iv). Apply the construction of Lemma 3 to c-graph $\mathcal{C}^+_{con}(H^+_i,\mathcal{T}^+_i)$ and cycle ρ to obtain a c-graph $\mathcal{C}^+(S^+_i,\mathcal{K}^+_i)$ satisfying Properties (2) and (3) of the lemma.
      ii. Augment $\mathcal{C}^+(S^+_i,\mathcal{K}^+_i)$ to a c-graph $\mathcal{C}^+_\Delta(R^+_i,\mathcal{J}^+_i)$ by introducing new vertices, each belonging to a distinct cluster, and by adding edges only between the vertices in $V(S^+_i)$ and the newly introduced vertices in such a way that cycle ρ bounds an inner face of R^+_i and all the other faces of R^+_i are triangles.
      iii. Merge $\mathcal{C}^+_\Delta(R^+_i,\mathcal{J}^+_i)$ and $\mathcal{C}^-_\rho(G^-,\mathcal{T}^-)$ to obtain a c-graph $\mathcal{C}^-(G^-_i,\mathcal{T}^-_i)$[1].
      iv. Run TESTCP($\mathcal{C}^-(G^-_i,\mathcal{T}^-_i)$) to test whether $\mathcal{C}^-(G^-_i,\mathcal{T}^-_i)$ is c-planar[2].
      v. If TESTCP($\mathcal{C}^-(G^-_i,\mathcal{T}^-_i)$) returns `YES`, then construct a c-planar c-connected super c-graph $C^*(G^*,\mathcal{T})$ of $\mathcal{C}(G,\mathcal{T})$ as follows. Let $\mathcal{C}^-_{con}(H^-_i,\mathcal{T}^-_i)$ be the c-planar c-connected c-graph returned by TESTCP($\mathcal{C}^-(G^-_i,\mathcal{T}^-_i)$). Remove all the vertices and edges of H^-_i in the exterior of cycle ρ, thus obtaining a new c-graph $\mathcal{C}_{in}(G_{in},\mathcal{T}_{in})$ in which cycle ρ bounds the outer face. Similarly, remove all the vertices and edges of H^+_i in the interior of cycle ρ, thus obtaining a new c-graph $\mathcal{C}_{out}(G_{out},\mathcal{T}_{out})$ in which cycle ρ bounds an inner face. Finally, merge $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$ to obtain c-graph $\mathcal{C}^*(G^*,\mathcal{T})$ and **return** `YES` along with c-graph $\mathcal{C}^*(G^*,\mathcal{T})$.
4. **return** `NO` if no c-graph $\mathcal{C}^-(G^-_i,\mathcal{T}^-_i)$, constructed at step 3(a)iii, is c-planar.

Claim 1. *C-graph $\mathcal{C}^*(G^*, \mathcal{T})$ is a c-planar c-connected super c-graph of $\mathcal{C}(G, \mathcal{T})$.*

Proof. Graphs G_{in} and G_{out} are planar, as they are subgraphs of H_ω^- and H_ω^+, respectively (step 3(a)v). By construction, cycle ρ bounds an inner face of G_{out} and the outer face of G_{in}. Therefore G^*, obtained by merging G_{in} and G_{out}, is planar. Also, observe that, G_{in} and G_{out} are supergraphs of G^- and G^+, respectively, therefore graph G^* is a super graph of G.

We now show that $\mathcal{C}^*$ is c-connected, that is, for each cluster $\mu \in \mathcal{T}$, graph $G^*(\mu)$ is connected.

First, let μ be a cluster in $\mathcal{T}$ such that $V(\mu)$ lies in the interior of ρ in G. Since $\mathcal{C}^-_{con}(H_\omega^-, \mathcal{T}_\omega^-)$ is c-connected, we have that $H_\omega^-(\mu)$ is connected. Also, $V(\mu)$ lie in the interior of ρ in H_ω^-. By construction, G_{in} contains all the vertices and the edges in the interior of ρ, therefore we also have that $G_{in}(\mu)$ is connected. Hence, $G^*(\mu)$ is connected. The proof that graph $G^*(\mu)$ is connected, for each cluster μ in $\mathcal{T}$ such that $V(\mu)$ lies in the exterior of ρ in G, is analogous.

Then, let μ be a cluster such that $V(\mu) \cap V(\rho) \neq \emptyset$. Clearly, if $V(\mu) \subseteq V(\rho)$, then $G^*(\mu)$ is connected since both $G_{in}(\mu)$ and $G_{out}(\mu)$ are connected. Otherwise, the following three cases are possible: either $G_{in}(\mu)$ is disconnected, or $G_{out}(\mu)$ is disconnected, or both $G_{in}(\mu)$ and $G_{out}(\mu)$ are disconnected.

We show that all the vertices in $G_{in}(\mu)$ and in $G_{out}(\mu)$ are connected in $G^*(\mu)$.

We first prove that all the vertices in $G_{in}(\mu)$ are connected in $G^*(\mu)$.

Consider two connected components c' and c'' of $G_{in}(\mu)$. Observe that, by construction, c-graph $\mathcal{C}^-_{con}(H_\omega^-, \mathcal{T}_\omega^-)$ (step 3(a)v) is a merge of $\mathcal{C}_{in}(G_{in}, \mathcal{T}_{in})$ and of $\mathcal{C}^+_\Delta(R_\omega^+, \mathcal{J}_\omega^+)$. Since $\mathcal{C}^-_{con}$ is c-connected and since R_ω^+ is an augmentation of cycle-star S_ω^+ such that edges in $E(R_\omega^+) \setminus E(S_\omega^+)$ do not have endpoints in the same cluster, the c-graph $\mathcal{C}^\#(G^\#, \mathcal{T}^\#)$ obtained by merging $\mathcal{C}_{in}$ and $\mathcal{C}^+(S_\omega^+, \mathcal{K}_\omega^+)$ is also c-connected. Since $\mathcal{C}^\#$ is c-connected, the vertices of c' and c'' are connected via star vertices of S_ω^+ and vertices of G_{in} belonging to cluster μ in $G^\#(\mu)$. Observe that, by construction, c-graph $\mathcal{C}^+_{con}(H_\omega^+, \mathcal{T}_\omega^+)$ is a merge of $\mathcal{C}_{out}(G_{out}, \mathcal{T}_{out})$ and of $\mathcal{C}^-_\Delta(R_\omega, \mathcal{J}_\omega)$. Further, S_ω^+ has been obtained by applying the construction of Lemma 3 to c-graph $\mathcal{C}^+_{con}(H_\omega^+, \mathcal{T}_\omega^+)$ (step 3(a)i) and cycle ρ. Therefore, each connected component of μ in G_{out} corresponds to a star vertex of S_ω^+. Hence, we have that the vertices of c' and c'' are also connected in G^* via vertices of G_{out} and G_{in} belonging to cluster μ.

Now, we prove that all the vertices in $G_{out}(\mu)$ are connected in $G^*(\mu)$.

Consider two connected components c' and c'' of $G_{out}(\mu)$. Observe that, as shown above, each connected component of μ in G_{out} corresponds to a star vertex of S_ω^+. Recall that $\mathcal{C}^\#$ is c-connected. Therefore, the star vertices of S_ω^+ corresponding to c' and c'' are connected via other star vertices of S_ω^+ and vertices of G_{in} belonging to cluster μ in $G^\#(\mu)$. Hence, the vertices of c' and c'' are also connected in G^* via vertices of G_{out} belonging to connected components of μ corresponding to star vertices of S_ω^+ and vertices of G_{in} belonging to cluster μ in $G^*(\mu)$. □

($\Leftarrow$) Suppose that $\mathcal{C}(G, \mathcal{T})$ is c-planar. We show that Procedure OuterCheck and InnerCheck succeed. Since $\mathcal{C}(G, \mathcal{T})$ is c-planar, there exists a super c-graph $\mathcal{C}^*(G^*, \mathcal{T})$ of $\mathcal{C}$ such that G^* is planar and $\mathcal{C}^*$ is c-connected, by Theorem 2. By using the construction of Lemma 3 on c-graph $\mathcal{C}^*$, we can obtain a cycle-star S^- whose universal cycle is ρ that represents the connectivity of clusters inside ρ in $\mathcal{C}^*$. The proof of this direction follows from the next claim.

Claim 2. *Procedures* OuterCheck *and* InnerCheck *succeed for $S_i^- = S^-$.*

Proof. Procedure OuterCheck succeeds if, for a cycle separator ρ of G selected at step 1 of the algorithm, there exists a cycle-star S_i^- whose universal cycle is ρ such that the corresponding c-graph $\mathcal{C}^+(G_i^+, \mathcal{T}_i^+)$, constructed at steps 2(b)i, 2(b)ii, and 2(b)iii of the algorithm, is c-planar. Recall that, cycle-star S^- has the following properties: 1. Cycle ρ is the universal cycle of S^- and bounds the outer face of S^-, and 2. for each star vertex v of S^-, the neighbours of v belong to the same cluster $\mu \in \mathcal{K}^-$vertex v belongs to. Since, steps 2a and 2(b)i construct all c-graphs $\mathcal{C}^-(S_i^-, \mathcal{K}_i^-)$ with the above properties, when $S_i^- = S^-$ we are guaranteed to compute c-graph $\mathcal{C}^-(S^-, \mathcal{K}^-)$. First, observe that the c-graph obtained by merging c-graphs $\mathcal{C}^-(S^-, \mathcal{K}^-)$ and $\mathcal{C}^+_\rho(G^+, \mathcal{T}^+)$ is c-planar, since S^- has been obtained by applying the construction of Lemma 3 to a super c-connected c-planar c-graph $\mathcal{C}^*$ of $\mathcal{C}$. This together with Remark 1 imply that c-graph $\mathcal{C}^+(G_i^+, \mathcal{T}_i^+)$ is c-planar. Thus, the invocation of TestCP on $\mathcal{C}^+(G_i^+, \mathcal{T}_i^+)$ at step 2(b)iv will return `YES`. Hence, Procedure OuterCheck succeeds.

Procedure InnerCheck succeeds if, there exists a c-graph $\mathcal{C}^-(G_i^-, \mathcal{T}_i^-)$, constructed at steps 3(a)i, 3(a)ii, and 3(a)iii of the algorithm, that is c-planar. By Theorem 2, a c-graph $\mathcal{C}^-(G_i^-, \mathcal{T}_i^-)$ is c-planar if and only if there exists a super c-graph $\mathcal{C}'(G', \mathcal{T}_i^-)$ of $\mathcal{C}^-(G_i^-, \mathcal{T}_i^-)$ such that G' is planar and $\mathcal{C}'$ is c-connected. As Procedure OuterCheck succeeds, the c-graph $\mathcal{C}^+(G_i^+, \mathcal{T}_i^+)$ corresponding to S^- is c-planar. Therefore, Procedure OuterCheck provides us with a c-planar c-connected c-graph $\mathcal{C}^+_{con}(H_i^+, \mathcal{T}_i^+)$ (see steps 2(b)iv and 3(a)i) that is a super c-graph of $\mathcal{C}^+(G_i^+, \mathcal{T}_i^+)$. Consider the c-graph $\mathcal{C}^+(S_i^+, \mathcal{K}_i^+)$ constructed at step 3(a)i by applying the construction of Lemma 3 to $\mathcal{C}^+_{con}$. Observe that, the c-graph obtained by merging $\mathcal{C}^+(S_i^+, \mathcal{K}_i^+)$ and $\mathcal{C}^-_\Delta(R_i^-, \mathcal{J}_i^-)$ is a c-connected c-planar c-graph. This is due to the fact that, since R_i^- is internally triangulated, there exists no edge in the interior of ρ in H_i^+ that belongs to H_i^+ and does not belong to R_i^-, that is, no candidate saturating edges connect two vertices in the interior of ρ in $\mathcal{C}^+_{con}$. Since $S_i^+ \subseteq R_i^+$, we also have that the c-graph obtained by merging $\mathcal{C}^+_\Delta(R_i^+, \mathcal{J}_i^+)$ (constructed at step 3(a)ii) and $\mathcal{C}^-_\Delta(R_i^-, \mathcal{J}_i^-)$ is a c-connected c-planar c-graph. Also, since each of the vertices added to obtain R_i^- from S^- belongs to a different cluster and since the edges added to internally triangulate S^- do not connect vertices of the same cluster, we have that the c-graph obtained by merging $\mathcal{C}^+_\Delta(R_i^+, \mathcal{J}_i^+)$ and $\mathcal{C}^-(S^-, \mathcal{K}^-)$ is also a c-connected c-planar c-graph.

Let $\mathcal{A}$ be the subgraph of G^* induced by the edges in the interior and on the boundary of ρ in $\mathcal{C}^*$. Since S^- exactly represents the cluster connectivity of $\mathcal{A}$, the c-graph obtained by merging $\mathcal{C}^+_\Delta(R_i^+, \mathcal{J}_i^+)$ and $\mathcal{A}$ is also a c-connected c-planar c-graph. The fact that, such a c-graph is a super c-graph of $\mathcal{C}^-(G_i^-, \mathcal{T}_i^-)$ shows that $\mathcal{C}^-(G_i^-, \mathcal{T}_i^-)$ is c-planar. Hence, Procedure InnerCheck succeeds. □

We are finally ready to present the main result of the section.

Theorem 3. *The* C-Planarity *problem can be solved in* $2^{O(\sqrt{\ell n}\cdot\log n)}$ *time for* n*-vertex c-graphs with maximum face size* ℓ.

Proof. Given an n-vertex c-graph $\mathcal{C}(G, \mathcal{T})$ with maximum face size ℓ, by Lemma 2, we can construct in linear time a 2-connected, in fact 3-connected, c-graph $\mathcal{C}'$ equivalent to $\mathcal{C}$. Therefore, we can apply Algorithm TestCP to $\mathcal{C}'$ to determine whether $\mathcal{C}$ is c-planar.

We now argue about the running time.

Since G' is 2-connected and since, by Lemma 2, $|V(G')| = O(|V(G)|)$ and the maximum face size ℓ' of G' is $O(\ell)$, we can construct a cycle separator ρ of G of size $s(n) = O(\sqrt{\ell n})$ that separates the vertices in the interior of ρ from the vertices in the exterior of ρ in such a way that both such sets contain at most $\frac{2n}{3}$ vertices [33]. Also, since all cycle-stars whose universal cycle is ρ have size $O(s(n))$ and the augmentations at steps 2(b)ii and 3(a)ii can be done by introducing at most $s(n)$ new vertices, graphs G_i^+ (step 2(b)iv) and G_i^- (step 3(a)iv) have $O(\frac{2n}{3} + O(s(n)))$ size. Further, by construction, G_i^- and G_i^+ are 2-connected and their maximum face size is ℓ'; thus, the cycle separators of G_i^- and G_i^+ have size bounded by $s(|V(G_i^-)|)$ and by $s(|V(G_i^+)|)$, respectively.

Moreover, observe that each cycle-star $S_i^- \in \mathcal{S}$ satisfying the properties described at step 2a can be constructed in $O(s(n))$ time. Also, each cycle-star S_i^- is in one-to-one correspondence with a non-crossing partition of a set containing $s(n)$ elements. This is due to the fact that each vertex of ρ is incident to at most a star vertex of S_i^- and that, by the planarity of S_i^-, the neighbours of any two star vertices do not alternate along ρ. The number of all such partitions is expressed by the Catalan number $C_{s(n)} \leq 4^{s(n)}$.

The non-recursive running time $f(n)$ is bounded by the time taken by steps 1 and 3(a)i, that is, $O(n)$ time. In fact, the cycle-separator of an n-vertex graph can constructed in $O(n)$ time [33]. Testing whether a cycle is a cluster-separator can be done by performing a visit of the graph to detect if there exist a cluster whose vertices lie inside and outside of ρ, but not along ρ; this can clearly be done in $O(n)$ time. Finally, applying the construction of Lemma 3 to obtain a cycle-star only requires finding the connected components of each cluster inside (or outside) ρ and their respective connections to cycle ρ, which can be done in $O(n)$ time by performing a DFS-visit of G^- (or G^+).

By the above arguments, the running time of Algorithm TestCP is expressed by by the following recurrence:

$$T(n) = 2C_{s(n)}\Big(T\Big(\frac{2n}{3} + O\big(s(n)\big)\Big) + f(n)\Big) \tag{1}$$

Since equation (1) solves to $T(n) = 2^{O(\sqrt{\ell n}\cdot\log n)}$ for $s(n) = O(\sqrt{\ell n})$, $C_{s(n)} = 4^{s(n)}$, $f(n) = O(n)$, the statement follows. □

In the next section, we show how to adapt algorithm TestCP to obtain an XP algorithm with parameter h for generalized h-simply nested graphs, which extend simply-nested graphs with bounded face size.

3.1 Generalized h-Simply-Nested Graphs

A plane graph is *h-simply-nested* if it consists of nested cycles of size at most h and of edges only connecting vertices of the same cycle or vertices of adjacent cycles; refer to Fig. 6. We extend the class of h-simply-nested graphs to the class of *generalized h-simply-nested graphs*, by allowing the inner-most cycle to contain a plane graph consisting of at most $2h$ vertices in its interior and the outer-most cycle to contain a plane graph consisting of at most $2h$ vertices in its exterior. See [16] for a related graph class, in which the vertices in the interior of the inner-most cycle can only form a tree, there exist no other vertices in the exterior of the outer-most cycle, and chords are not allowed for the remaining cycles.

Let G be a generalized h-simply-nested plane graph with $n > 5h$ vertices. We have the following simple observation about the structure of G; refer to Fig. 6.

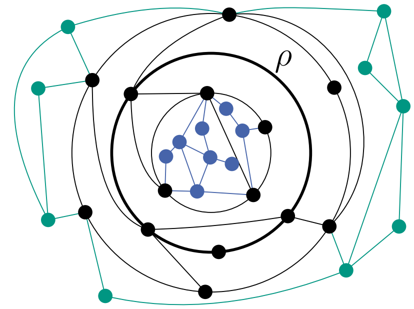


Fig. 6: A generalized 6-simply-nested graph.

Observation 2. *Graph G contains a cycle ρ with $|V(\rho)| \leq h$ that separates G into two generalized h-simply-nested graphs G^+ and G^- with $|V(G^+)| \leq \frac{n}{2}$ and $|V(G^-)| \leq \frac{n}{2}$ such that G^+ (G^-) does not contain any vertex in the exterior (interior) of its outer-most cycle (inner-most cycle). Further, such a cycle can be computed in $O(n)$ time.*

By Observation 2, we can use a cycle separator of size at most h in Algorithm TestCP to test the c-planarity of a c-graph whose underlying graph is a generalized h-simply-nested plane graph G (instead of a cycle separator of size $O(\sqrt{\ell n})$, where ℓ is the maximum face size of G). Observe that, graphs G_i^+ and G_i^- obtained at steps 2(b)iii and 2(b)iii of the algorithm, respectively, also belong to the family of generalized h-simply-nested plane graphs. Therefore, Observation 2 also holds for such graphs. Altogether, we obtain the following recurrence relation for the running-time:

$$T(n) = 2C_h\left(T\left(\frac{n}{2} + O(h)\right) + O(n)\right) \tag{2}$$

Equation (2) immediately implies the following theorem.

Theorem 4. *The* C-Planarity *problem can be solved in $n^{O(h)}$ time for n-vertex c-graphs whose underlying graph is a generalized h-simply-nested graph.*

4 An MSO_2 formulation for C-Planarity

In this section, we show that the property of a c-graph of admitting a c-planar drawing can be expressed in extended monadic second-order (MSO_2) logic. We use this result and the fact that graph properties definable in MSO_2 logic can be verified in linear time on graphs of bounded treewidth, by Courcelle's Theorem [19], to build an FPT algorithm for testing the c-planarity of embedded flat c-graphs.

First-order graph logic deals with formulae whose variables represent the vertices and edges of a graph. *Second-order graph logic* also allows quantification over k-ary relations defined on the vertices and edges. *MSO_2 logic* only allows quantification over elements and *unary relations*, that is, sets of vertices and edges. Given a graph G and an MSO_2 formula ϕ, we say that G *models* ϕ, denoted by $G \models \phi$, if the logic statement expressed by ϕ is satisfied by the vertices, edges, and sets of vertices and edges in G. We will apply Courcelle's theorem not to the underlying graph G of the clustered planarity instance, but to the supergraph $G^\diamond$ of G that includes all the candidate saturating edges of G. This will allow us to quantify over sets of candidate saturating edges, but in exchange we must show that $G^\diamond$, and not just G, has low treewidth (Lemma 2).

Let H be a graph and let $E_1, E_2 \subseteq E(H)$. The following logic predicates can be expressed in MSO_2 logic (refer, e.g., to [8,20] for their detailed formulation):

- $\diamond$ $\textsc{planar}_H(E_1, E_2)$:= the subgraph $(V(H), E_1 \cup E_2)$ of H is planar, and
- $\diamond$ $\textsc{conn}_H(U, E_1, E_2)$:= vertices in $U \subseteq V(H)$ are connected by edges in $E_1 \cup E_2$.

Let $\mathcal{C}(G, \mathcal{T})$ be a c-graph and let E^* be the set of all the candidate saturating edges of $\mathcal{C}$. By Property(iii) of Theorem 2, c-graph $\mathcal{C}$ admits a c-planar drawing if and only if there exists a super c-graph $\mathcal{C}'(G', \mathcal{T})$ of $\mathcal{C}$ such that G' is planar and $\mathcal{C}'$ is c-connected. Testing Property(iii) amounts to determining the existence of a set $E^+ \subseteq E^*$ such that (i) the subgraph G' of $G^\diamond$ obtained by adding the edges in E^+ to G is planar and (ii) graph $G'(\mu)$ is connected, for each cluster $\mu \in \mathcal{T}$.

We remark that in an MSO_2 formula it is possible to refer to given subsets of vertices or edges of a graph, provided that the elements of such subsets can be distinguished from the elements of other subsets by equipping them with labels from a constant finite set [6]. Therefore, in our formulae we use the unquantified variables V_i to denote the set of vertices belonging to cluster μ_i, for each disconnected cluster $\mu_i \in \mathcal{T}$, E^* to denote the set consisting of all the candidate saturating edges of $\mathcal{C}$, and E_G to denote $E(G)$.

Let c be the number of disconnected clusters in $\mathcal{T}$. We have the formula:

$$\textsc{c-planar}_{\mathcal{C}(G,\mathcal{T})} \equiv \exists(E^+ \subseteq E^*)\Big[\textsc{planar}_{G^\diamond}(E_G, E^+) \wedge \bigwedge_{i=1}^{c} \textsc{conn}_{G^\diamond}(V_i, E_G, E^+)\Big]$$

It is easy to see that formula $\textsc{c-planar}_{\mathcal{C}(G,\mathcal{T})}$ correctly expresses Condition(iii) of Theorem 2 only if G admits a unique combinatorial embedding (up to a flip). In fact, if G has more than one embedding, formula $\textsc{c-planar}_{\mathcal{C}(G,\mathcal{T})}$ might still be satisfiable after a change of the embedding, as formula $\textsc{planar}_{G^\diamond}(E_G, E^+)$ models the planarity of an abstract graph rather than the planarity of a combinatorial embedding. We formalize this fact in the following lemma.

Lemma 4. *Let $\mathcal{C}(G,\mathcal{T})$ be a c-graph such that G has a unique combinatorial embedding and let $\mathcal{C}^\diamond(G^\diamond,\mathcal{T}^\diamond)$ be the c-graph obtained by augmenting $\mathcal{C}$ with all its candidate saturating edges. Then, $\mathcal{C}$ is c-planar iff $G^\diamond \models \textsc{c-planar}_{\mathcal{C}(G,\mathcal{T})}$.*

Since changes of embedding are not allowed in our context, as we aim at testing the c-planarity of a c-graph given a prescribed embedding, we combine Lemmata 2 and 4, and then invoke Courcelle's Theorem to obtain the following main result.

Theorem 5. *The* C-Planarity *problem can be solved in $f(\overline{\text{emw}}, c)O(n)$ time for n-vertex c-graphs with c disconnected clusters and whose underlying graph has embedded-width $\overline{\text{emw}}$, where f is a computable function.*

Proof. To test that $\mathcal{C}(G,\mathcal{T})$ admits a c-planar drawing with the given embedding we proceed as follows. First, we apply Lemma 2 to obtain a c-graph $\mathcal{C}^*(G^*,\mathcal{T}^*)$ that is equivalent to $\mathcal{C}(G,\mathcal{T})$ such that G^* is 3-connected. Note that, the 3-connectivity of G^* implies that it has a unique combinatorial embedding (up to a flip) [34]. Then, we construct formula $\phi = \textsc{c-planar}_{\mathcal{C}^*(G^*,\mathcal{T}^*)}$ and the super c-graph $\mathcal{C}^\diamond(G^\diamond,\mathcal{T}^\diamond)$ of $\mathcal{C}^*$ obtained by augmenting $\mathcal{C}^*$ with all its candidate saturating edges. Finally, we use Courcelle's Theorem to test whether $G^\diamond \models \phi$. The correctness immediately follows from Lemmata 2 and 4.

We now argue about the running time. By Lemma 2, c-graph $\mathcal{C}^*(G^*,\mathcal{T}^*)$ can be constructed in $O(n)$ time. Let κ be the maximum face size of G^*. The number of candidate saturating edges of $\mathcal{C}^*$ is $O(\kappa^2 n)$. By Lemma 2, $\kappa = O(\ell)$. Hence, we can augment $\mathcal{C}^*(G^*,\mathcal{T}^*)$ to obtain $\mathcal{C}^\diamond(G^\diamond,\mathcal{T}^\diamond)$ in $O(\ell^2 n)$ time.

By Courcelle's theorem [19], it is possible to verify whether $G^\diamond \models \phi$ in $g(tw(G^\diamond), len(\phi))O(|V(G^\diamond)| + |E(G^\diamond)|)$ time, where g is a computable function. By Lemma 2, $|V(G^\diamond)| = |V(G^*)| = O(n)$ and $\text{tw}(G^\diamond) = \overline{\text{emw}}(G)$. Also, by the discussion above, $|E(G^\diamond)| = O(\ell^2 n)$ and, by definition of embedded-width, $\ell = O(\overline{\text{emw}})$; thus, $|E(G^\diamond)| = O(\overline{\text{emw}}^2 n)$. Further, formula ϕ can be constructed in time proportional to its length $len(\phi)$, which is $O(c)$. Therefore, the overall running time can be expressed as $f(\overline{\text{emw}}, c)O(n)$, where f is a computable function. ◻

5 Conclusions and Open Problems

In this paper, we provide subexponential-time, XP, and FPT algorithms to test C-Planarity of fairly-broad classes of c-graphs.

Several interesting questions arise from this research: (1) Can our results be generalized from flat to non-flat c-graphs? (2) Is there a fully polynomial-time algorithm to test C-Planarity of c-graphs whose underlying graph is a generalized h-simply-nested graph? (3) Are there interesting parameters of the underlying graph such that C-Planarity is FPT with respect to a single one of them (e.g., outerplanarity index, maximum face size, notable graph width parameters)? (4) Are there interesting parameters of the c-graph such that C-Planarity is FPT with respect to a single one of them (e.g., number of clusters, number of vertices of the same cluster incident to the same face[3], maximum distance between two faces containing vertices of the same cluster)?

[3] This question has also been previously asked by Chimani *et al.* [13]

References

1. Akitaya, H.A., Fulek, R., Tóth, C.D.: Recognizing weak embeddings of graphs. In: Czumaj, A. (ed.) Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018. pp. 274–292. SIAM (2018), https://doi.org/10.1137/1.9781611975031.20
2. Angelini, P., Da Lozzo, G.: Clustered planarity with pipes. In: Hong, S. (ed.) 27th International Symposium on Algorithms and Computation, ISAAC 2016. LIPIcs, vol. 64, pp. 13:1–13:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016), https://doi.org/10.4230/LIPIcs.ISAAC.2016.13
3. Angelini, P., Da Lozzo, G., Di Battista, G., Frati, F.: Strip planarity testing for embedded planar graphs. Algorithmica 77(4), 1022–1059 (2017), https://doi.org/10.1007/s00453-016-0128-9
4. Angelini, P., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M., Roselli, V.: Relaxing the constraints of clustered planarity. Comput. Geom. 48(2), 42–75 (2015), https://doi.org/10.1016/j.comgeo.2014.08.001
5. Angelini, P., Frati, F., Patrignani, M.: Splitting clusters to get c-planarity. In: Eppstein, D., Gansner, E.R. (eds.) Graph Drawing, 17th International Symposium, GD 2009, Revised Papers. LNCS, vol. 5849, pp. 57–68. Springer (2009), https://doi.org/10.1007/978-3-642-11805-0_8
6. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. J. Algorithms 12(2), 308–340 (1991), https://doi.org/10.1016/0196-6774(91)90006-K
7. Athenstädt, J.C., Cornelsen, S.: Planarity of overlapping clusterings including unions of two partitions. J. Graph Algorithms Appl. 21(6), 1057–1089 (2017), https://doi.org/10.7155/jgaa.00450
8. Bannister, M.J., Eppstein, D.: Crossing minimization for 1-page and 2-page drawings of graphs with bounded treewidth. In: Duncan, C.A., Symvonis, A. (eds.) GD 2014. LNCS, vol. 8871, pp. 210–221. Springer (2014), https://doi.org/10.1007/978-3-662-45803-7_18
9. Biedl, T.: Drawing planar partitions III: Two Constrained Embedding Problems. Tech. Report RRR 13-98, Rutcor Research Report (1998)
10. Bläsius, T., Rutter, I.: A new perspective on clustered planarity as a combinatorial embedding problem. Theor. Comput. Sci. 609, 306–315 (2016), https://doi.org/10.1016/j.tcs.2015.10.011
11. Borradaile, G., Erickson, J., Le, H., Weber, R.: Embedded-width: A variant of treewidth for plane graphs (2017), http://arxiv.org/abs/1703.07532
12. Bouchitté, V., Mazoit, F., Todinca, I.: Treewidth of planar graphs: connections with duality. Electronic Notes in Discrete Mathematics 10, 34–38 (2001), https://doi.org/10.1016/S1571-0653(04)00353-1
13. Chimani, M., Di Battista, G., Frati, F., Klein, K.: Advances on testing c-planarity of embedded flat clustered graphs. In: Duncan, C.A., Symvonis, A. (eds.) Graph Drawing - 22nd International Symposium, GD 2014, Revised Selected Papers. LNCS, vol. 8871, pp. 416–427. Springer (2014), https://doi.org/10.1007/978-3-662-45803-7_35
14. Chimani, M., Gutwenger, C., Jansen, M., Klein, K., Mutzel, P.: Computing maximum c-planar subgraphs. In: Tollis, I.G., Patrignani, M. (eds.) Graph Drawing, 16th International Symposium, GD 2008, Revised Papers. LNCS, vol. 5417, pp. 114–120. Springer (2008), https://doi.org/10.1007/978-3-642-00219-9_12
15. Chimani, M., Klein, K.: Shrinking the search space for clustered planarity. In: Didimo, W., Patrignani, M. (eds.) Graph Drawing - 20th International Symposium, GD 2012, Revised Selected Papers. LNCS, vol. 7704, pp. 90–101. Springer (2012), https://doi.org/10.1007/978-3-642-36763-2_9
16. Cimikowski, R.J.: Finding hamiltonian cycles in certain planar graphs. Inf. Process. Lett. 35(5), 249–254 (1990), https://doi.org/10.1016/0020-0190(90)90053-Z
17. Cornelsen, S., Wagner, D.: Completely connected clustered graphs. J. Discrete Algorithms 4(2), 313–323 (2006), https://doi.org/10.1016/j.jda.2005.06.002
18. Cortese, P.F., Di Battista, G., Frati, F., Patrignani, M., Pizzonia, M.: C-planarity of c-connected clustered graphs. J. Graph Algorithms Appl. 12(2), 225–262 (2008), http://jgaa.info/accepted/2008/Cortese+2008.12.2.pdf
19. Courcelle, B.: The monadic second-order logic of graphs. i. recognizable sets of finite graphs. Inf. Comput. 85(1), 12–75 (1990), https://doi.org/10.1016/0890-5401(90)90043-H
20. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015), https://doi.org/10.1007/978-3-319-21275-3
21. Dahlhaus, E.: A linear time algorithm to recognize clustered graphs and its parallelization. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN '98: Theoretical Informatics, Third Latin American Symposium, Proceedings. LNCS, vol. 1380, pp. 239–248. Springer (1998), https://doi.org/10.1007/BFb0054325
22. Di Battista, G., Frati, F.: Efficient c-planarity testing for embedded flat clustered graphs with small faces. J. Graph Algorithms Appl. 13(3), 349–378 (2009), http://jgaa.info/accepted/2009/DiBattistaFrati2009.13.3.pdf
23. Didimo, W., Giordano, F., Liotta, G.: Overlapping cluster planarity. J. Graph Algorithms Appl. 12(3), 267–291 (2008), http://jgaa.info/accepted/2008/DidimoGiordanoLiotta2008.12.3.pdf
24. Feng, Q., Cohen, R.F., Eades, P.: Planarity for clustered graphs. In: Spirakis, P.G. (ed.) Algorithms - ESA '95, Third Annual European Symposium, Proceedings. LNCS, vol. 979, pp. 213–226. Springer (1995), https://doi.org/10.1007/3-540-60313-1_145
25. Fulek, R.: C-planarity of embedded cyclic c-graphs. In: Hu, Y., Nöllenburg, M. (eds.) Graph Drawing. pp. 94–106. LNCS 9801 (2016)

26. Fulek, R., Kyncl, J., Malinovic, I., Pálvölgyi, D.: Clustered planarity testing revisited. Electr. J. Comb. 22(4), P4.24 (2015), http://www.combinatorics.org/ojs/index.php/eljc/article/view/v22i4p24
27. Goodrich, M.T., Lueker, G.S., Sun, J.Z.: C-planarity of extrovert clustered graphs. In: Healy, P., Nikolov, N.S. (eds.) Graph Drawing, 13th International Symposium, GD 2005, Revised Papers. LNCS, vol. 3843, pp. 211–222. Springer (2005), https://doi.org/10.1007/11618058_20
28. Gutwenger, C., Jünger, M., Leipert, S., Mutzel, P., Percan, M., Weiskircher, R.: Advances in c-planarity testing of clustered graphs. In: Kobourov, S.G., Goodrich, M.T. (eds.) Graph Drawing, 10th International Symposium, GD 2002, Revised Papers. LNCS, vol. 2528, pp. 220–235. Springer (2002), https://doi.org/10.1007/3-540-36151-0_21
29. Hong, S.H., Nagamochi, H.: Simpler algorithms for testing two-page book embedding of partitioned graphs. Theoretical Computer Science (2016), http://www.sciencedirect.com/science/article/pii/S0304397515012207
30. Jelínek, V., Jelínková, E., Kratochvíl, J., Lidický, B.: Clustered planarity: Embedded clustered graphs with two-component clusters. In: Tollis, I.G., Patrignani, M. (eds.) Graph Drawing, 16th International Symposium, GD 2008, Revised Papers. LNCS, vol. 5417, pp. 121–132. Springer (2008), https://doi.org/10.1007/978-3-642-00219-9_13
31. Jelínková, E., Kára, J., Kratochvíl, J., Pergel, M., Suchý, O., Vyskocil, T.: Clustered planarity: Small clusters in cycles and eulerian graphs. J. Graph Algorithms Appl. 13(3), 379–422 (2009), http://jgaa.info/accepted/2009/Jelinkova+2009.13.3.pdf
32. Lengauer, T.: Hierarchical planarity testing algorithms. J. ACM 36(3), 474–509 (1989), http://doi.acm.org/10.1145/65950.65952
33. Miller, G.L.: Finding small simple cycle separators for 2-connected planar graphs. J. Comput. Syst. Sci. 32(3), 265–279 (1986), https://doi.org/10.1016/0022-0000(86)90030-9
34. Whitney, H.: Congruent graphs and the connectivity of graphs. American Journal of Mathematics 54(1), 150–168 (1932), http://www.jstor.org/stable/2371086