Planarity of Overlapping Clusterings Including Unions of Two Partitions

Jan C. Athenstädt  Sabine Cornelsen

Department of Computer and Information Science, University of Konstanz

Abstract

We consider clustered planarity with overlapping clusters as introduced by Didimo et al. [14]. It can be deduced from a proof in Athenstädt et al. [2] that the problem is NP-complete, even if restricted to instances where the underlying graph is 2-connected, the set of clusters is the union of two partitions and each cluster contains at most two connected components while their complements contain at most three connected components.

In this paper, we show that clustered planarity with overlapping clusters can be solved in polynomial time if each cluster induces a connected subgraph. It can be solved in linear time if the set of clusters is the union of two partitions of the vertex set such that, for each cluster, both the cluster and its complement induce connected subgraphs.

This research has received funding from the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no. 319209.

E-mail addresses: jan.athenstaedt@uni-konstanz.de (Jan C. Athenstädt) sabine.cornelsen@uni-konstanz.de (Sabine Cornelsen)
1 Introduction

An (overlapping) clustered graph \( (G = (V, E), \mathcal{C}) \) consists of an undirected graph \( G \) and a set \( \mathcal{C} \) of subsets of the vertex set \( V \). The elements of \( \mathcal{C} \) are called clusters. A vertex may be contained in several clusters. Moreover, clusters may overlap, i.e., there might be \( C_1, C_2 \in \mathcal{C} \) with \( C_1 \cap C_2 \neq \emptyset \), \( C_1 \not\subseteq C_2 \), and \( C_2 \not\subseteq C_1 \).

Observe that the abstract Hasse diagram of \( \mathcal{C} \cup \{\{v\}, v \in V\} \cup \{V\} \) is a succinct representation of an overlapping clustering \( \mathcal{C} \) — actually, Didimo et al. [14] define overlapping clustered graphs via the Hasse diagram.

Didimo et al. [14] defined planarity for overlapping clustered graphs geometrically: An overlapping clustered graph \( (G = (V, E), \mathcal{C}) \) is clustered planar if the vertices can be represented by distinct points, each edge \( e \in E \) by a curve \( R(e) \) between its incident vertices, and each cluster \( C \in \mathcal{C} \) by a simple closed region \( R(C) \) in the plane such that for \( X, Y \in E \cup \mathcal{C} \) we have that (i) \( X \subset R(X) \), \( (V \setminus X) \cap R(X) = \emptyset \), (ii) \( R(X) \subseteq R(Y) \) if \( X \subseteq Y \), and (iii) every connected region of \( R(X) \cap R(Y) \) contains a vertex. (Recall that edges of an undirected graph are subsets of the vertex-set of size two.) Condition (i) means that edges may not contain non-incident vertices while clusters must enclose the vertices they contain and no others. Condition (ii) implies that cluster boundaries may not intersect if one cluster is contained in the other and that intra-cluster edges must be routed within the cluster. Condition (iii) ensures several things: (a) two edges intersect at most in a common incident vertex, (b) an edge crosses a cluster boundary at most once, (c) if two clusters do not share a vertex, their regions must be disjoint (d) the regions of clusters with common vertices might intersect in several connected components, yet each of them must contain at least one vertex.

Thus, the clustered graph in Fig. 1a is clustered planar while the clustered graph in Fig. 1b is not.

Clustered planarity is NP-complete in general as shown in [17], where the case with \( E = \emptyset \) is examined. Didimo et al. [14] showed that clustered planarity can be solved in polynomial time for the special case where each cluster overlaps with at most one other cluster, each cluster as well as the intersection of any two clusters induce connected subgraphs, and some additional connectivity properties. They posed it as an open question whether clustered planarity is polynomial-time solvable for general overlapping clustered graphs under the only condition that each cluster induces a connected subgraph. We will answer this question in the affirmative.

If the clustering is hierarchical, i.e., if any two clusters in \( \mathcal{C} \) are either disjoint or one is contained in the other then clustered planarity is the classical problem of c-planarity as considered in [15]. One of the most important open problems in the field of Graph Drawing is the complexity of c-planarity of hierarchically clustered graphs. An overview on the classical c-planarity problem can be found in [5, 11, 19]. Dahlhaus [12] and later Cortese et al. [11] showed that c-planarity of hierarchically clustered graphs can be solved in linear time if each cluster...\footnote{For the sake of simplicity, we identify each vertex with its image in the plane.}
Figure 1: Two graphs with two clusters each (set of vertices enclosed by red dotted and blue dashed curve, respectively)

induces a connected subgraph. Their approaches make use of the decomposition of the graph into 3-connected components as represented by BC- and SPQR-trees.

Angelini et al. [1] defined drawings with region-region crossings of hierarchically clustered graphs. These are essentially representations by points and regions such that all conditions of clustered planarity are fulfilled except for Condition (iii) when $X$ and $Y$ are both clusters. As an example, Fig. 1b shows a drawing of a clustered graph with one region-region crossing. Observe that the intersection of the two regions does not contain a vertex as required by the definition of clustered planarity. Angelini et al. [1] showed how to use SPQR-trees to test in polynomial time whether any hierarchically clustered graph with an underlying 2-connected graph has a drawing with region-region crossings.

If $E = \emptyset$ then clustered planarity is closely related to the NP-complete problem of hypergraph (vertex) planarity as defined in [17]: Given a set $\mathcal{C}$ of subsets of a set $V$, is there a planar support, i.e., a planar graph $G = (V, E)$ such that each set in $\mathcal{C}$ induces a connected subgraph of $G$? Various subclasses of planar supports that directly imply clustered planarity – such as trees, cacti, and outerplanar supports – were considered [3, 4, 7, 8, 9, 18]. Hypergraph planarity remains NP-complete even if $\mathcal{C}$ is the union of two partitions [2].

The proof in [2] even shows that clustered planarity remains NP-complete if the underlying graph $G$ is 2-connected, $\mathcal{C}$ is the union of two partitions, each cluster contains at most two connected components, and the complement of any cluster contains at most three connected components.

**Contribution:** In Sect. 3 we focus on a special case: $\mathcal{C}$ is the union of two partitions and for each cluster, both the cluster itself and its complement are connected. Different from hierarchical clusterings, this connectivity property does not automatically imply clustered planarity in the overlapping case. Yet, for the special case considered, we can give a characterization that yields a linear-time testing algorithm. In Sect. 4 and 5 we show how to use BC-trees, SPQR-trees and the consecutive-ones property to obtain an algorithm for the more general case of testing clustered planarity of possibly overlapping, connected clusters. The run time of the algorithm is polynomial in $|V|$ and $|\mathcal{C}|$. 
Figure 2: The blue (dashed) and red (dotted) lines induce two partitions \( \mathcal{P}_B = \{ \{2\}, \{1,3,4,5\} \} \) and \( \mathcal{P}_R = \{ \{1,2,3\}, \{4,5\} \} \). This leads to an intersection partition \( \mathcal{P}_I = \{ \{2\}, \{1,3\}, \{4,5\} \} \) and a connected intersection partition \( \mathcal{P}'_I = \{ \{1\}, \{2\}, \{3\}, \{4,5\} \} \).

2 Preliminaries

For a subset \( C \subseteq V \) of the vertices of an undirected graph \( G = (V,E) \), we denote by \( G[C] \) the subgraph of \( G \) induced by \( C \), i.e. the graph with vertex set \( C \) and edge set \( \{ e \in E : e \subseteq C \} \). A \( C \)-path (\( C \)-cycle) is a path (simple cycle) in \( G[C] \).

For two subsets \( S,T \subseteq V \), let \( E(S,T) \) be the set of edges with one end vertex in \( S \) and the other end vertex in \( T \). A partition of \( V \) is a set \( \mathcal{P} \) of subsets of \( V \) such that each vertex in \( V \) is contained in exactly one set in \( \mathcal{P} \). For two partitions \( \mathcal{P}_B = \{ B_1, \ldots, B_{\ell_B} \} \) and \( \mathcal{P}_R = \{ R_1, \ldots, R_{\ell_R} \} \) of \( V \), we define the intersection partition \( \mathcal{P}_I = \{ B_i \cap R_j : i = 1, \ldots, \ell_B, j = 1, \ldots, \ell_R \} \). (This is also known as coarsest common refinement.) The connected intersection partition \( \mathcal{P}'_I \) of \( \mathcal{P}_B \) and \( \mathcal{P}_R \) is the partition induced by the connected components of \( G[C], C \in \mathcal{P}_I \), i.e. \( X \subseteq V \) is contained in \( \mathcal{P}'_I \) if and only if \( X \) is the set of vertices of a connected component of one of the graphs \( G[C], C \in \mathcal{P}_I \). See Fig. 2 for an example.

A consecutive-ones ordering of a binary matrix is a permutation of its columns such that in each row all of the 1s are consecutive, i.e. such that each row is of the form \( 0^*1^*0^* \). A binary matrix has the consecutive-ones property if and only if it has a consecutive-ones ordering. It can be tested in linear time whether a binary matrix has the consecutive-ones property and a consecutive-ones ordering can be found in linear time if it exists [6].

2.1 Planarity of Overlapping Clustered Graphs

Let \( (G = (V,E), \mathcal{C}) \) be an overlapping clustered graph. The clustered graph \( (G, \mathcal{C}) \) is c-connected if \( G[C] \) is connected for all \( C \in \mathcal{C} \) and c-co-connected if both, \( G[C] \) and \( G[V \setminus C] \), are connected for all \( C \in \mathcal{C} \).

If \( (G, \mathcal{C}) \) is c-connected then a c-planar embedding of \( G \) for \( \mathcal{C} \) is a planar embedding of \( G \) such that \( V \setminus C \) is in the outer face of \( G[C] \) for all \( C \in \mathcal{C} \). A graph \( G^+ = (V, E^+) \) is a c-planar support of a clustered graph \( (G = (V,E), \mathcal{C}) \) if \( E \subseteq E^+ \), \( (G^+, \mathcal{C}) \) is c-connected and there is a c-planar embedding of \( G^+ \) for \( \mathcal{C} \).
It was shown that a \( c \)-connected overlapping clustered graph \cite{14} or a hier-
archically clustered graph \cite{15}, respectively, is clustered planar in the sense of
\cite{14} if and only if it has a \( c \)-planar support.

In this work, we define any clustered graph to be \textit{\( c \)-planar} if and only if it
has a \( c \)-planar support.

### 2.2 BC-Trees

A vertex \( v \) is a \textit{cut vertex} of a connected graph \( G \) if the graph that results from
\( G \) by deleting \( v \) and its incident edges is not connected. A connected graph
is \textit{\( 2 \)-connected} if it contains more than two vertices but no cut vertices. The
\textit{blocks} of a connected graph are the maximally 2-connected subgraphs and the
subgraphs induced by bridges. The vertices of the \textit{block–cut tree (BC-tree)} of
a graph \( G \) are the blocks and the cut vertices of \( G \). There is an edge in the
block–cut tree between a block \( H \) and a cut vertex \( v \) if \( v \) is contained in \( H \). See
Fig. 16 for an illustration.

### 2.3 SPQR-Trees

A way of representing all possible embeddings of a 2-connected subgraph are
SPQR-trees. Two vertices \( v \) and \( w \) are a \textit{separation pair} of a 2-connected graph
\( G \) if the graph that results from \( G \) by deleting \( v \) and \( w \) and their incident edges
is not connected. A graph is \textit{\( 3 \)-connected} if it contains more than three vertices
but no separation pair. An \textit{SPQR-tree} \cite{13} is a labeled tree that represents the
decomposition of a 2-connected graph into 3-connected components. Each node
\( \nu \) of an SPQR-tree is labeled with a multi-graph \( \text{skel}(\nu) \) – called the \textit{skeleton}
of \( \nu \). There are four different types of labels associated with the skeletons: \textit{S-nodes}
for simple cycles, \textit{P-nodes} for three or more parallel edges, \textit{R-nodes} for a simple
3-connected graph, and \textit{Q-nodes} for two parallel edges.

The Q-nodes are the leaves of an SPQR-tree. Neither two S-nodes, nor two
P-nodes are adjacent in an SPQR-tree. For each node \( \nu \) of an SPQR-tree there
is a one-to-one correspondence of the edges of \( \text{skel}(\nu) \) and the edges incident to \( \nu \)
(except for the Q-nodes where one of the two edges of the skeleton corresponds
to the only incident edge of the Q-node). The edge of \( \text{skel}(\nu) \) corresponding to
the edge \( \{\nu, \mu\} \) of the SPQR-tree is denoted by \( e_\mu \). We consider the edges of
the skeletons oriented. For simplicity, we assume that the edges of the skeleton
of an S-node are oriented as a directed cycle and the edges of the skeleton of a
P-node are all oriented in parallel.

We consider the SPQR-tree \( T \) rooted at a Q-node \( r \). Let \( \nu \) be a node of \( T \).
The \textit{root edge} of \( \text{skel}(\nu) \) is the edge that corresponds to the parent edge of \( \nu \).
The \textit{poles} of \( \text{skel}(\nu) \) (or node \( \nu \), respectively) are the end vertices of the root
edge. Let \( \text{skel}^-(\nu) \) be the skeleton of \( \nu \) without the root edge.

Each node \( \nu \) of the rooted SPQR-tree represents a (multi-)graph \( G_r(\nu) \): The
Q-nodes (excluding the root) represent a graph with two vertices connected by
an edge and additionally by the root edge. Let \( \nu \) be a non-leaf node of an
SPQR-tree and let \( \nu_1, \ldots, \nu_k \) be the children of \( \nu \). For \( i = 1, \ldots, k \), remove the
edge associated with \( \{ \nu, \nu_1 \} \) from both \( \text{skel}(\nu) \) and \( \text{skel}(\nu_1) \). Insert the remaining parts of \( G_r(\nu_1) \) into \( \text{skel}(\nu) \) identifying the poles of \( G_r(\nu_1) \) with their counter parts in \( \text{skel}(\nu) \). The poles of \( G_r(\nu) \) are the poles of \( \nu \). Let \( G^-_r(\nu) \) be \( G_r(\nu) \) without the root edge of \( \text{skel}(\nu) \). (Some papers refer to \( G^-_r(\nu) \) as the pertinent graph.) \( G_r(r) \) is the graph represented by the SPQR-tree \( T \). The edges of \( G_r(r) \) correspond to the Q-nodes of the SPQR-tree. See Fig. 3 for an example.

If all skeletons have a planar embedding with the root edge on the outer face, then the construction yields a planar embedding of \( G_r(r) \) with the root edge on the outer face. On the other hand, any planar embedding of \( G_r(r) \) with the root edge on the outer face can be obtained, by permuting the ordering of the edges of the skeleton of a P-node, or by flipping the skeleton of an R-node around its root edge.

Every 2-connected graph is represented by a unique SPQR-tree (up to the choice of the root) and the SPQR-tree of a 2-connected graph can be constructed in linear time [10].

3 Two C-Co-Connected Partitions

In this section we show that c-planarity of a c-co-connected clustered graph can be tested in linear time if the set of clusters is the union of two partitions. Observe that in contrast to the hierarchical case [10], there are c-co-connected clustered graphs with an underlying planar graph that are not c-planar. See Fig. 4a for an example: The graph \( G = (V,E) \) is 3-connected and, thus, has a unique embedding up to the choice of the outer face. No matter which face we choose as the outer face, there is always at least one cluster \( C \) among the four clusters in \( P_B \cup P_R \) such that \( G[C] \) contains a simple cycle enclosing a vertex in \( V \setminus C \).

The key for the linear-time algorithm to test c-planarity of c-co-connected clustered graphs is the following characterization.

**Theorem 1** Let \( G = (V,E) \) be a graph and let \( P_R \) and \( P_B \) be two partitions of \( V \) such that the clustered graph \( (G,P_R \cup P_B) \) is c-co-connected. Let \( P'_R \) be the connected intersection partition of \( P_R \) and \( P_B \). Then \( (G,P'_R \cup P_B) \) is c-planar if and only if \( (G,P'_R) \) is c-planar.

**Proof:** We first show that if \( (G,P_B \cup P_R) \) is c-connected and c-planar then \( (G,P'_R) \) is c-planar: Consider a c-planar embedding of \( G \) for \( P_R \cup P_B \). Let \( C \in P'_R, R \in P_R, \) and \( B \in P_B \) with \( C \subseteq B \cap R \). Assume that there is a vertex \( v \) in an inner face of \( G[C] \). Observe that by c-planarity of \( (G,P_R \cup P_B) \), all vertices in the inner faces of \( G[C] \) are in \( B \cap R \). Since \( G[B] \) and \( G[R] \) are connected there must be a path from \( v \) to \( C \) that contains only vertices in one inner face of \( G[C] \) and thus in \( B \cap R \). Therefore, \( v \) and \( C \) are in the same connected component of \( G[B \cap R] \) and hence \( v \in C \). Thus no \( v \in V \setminus C \) is in an inner face of \( G[C] \) and therefore \( (G,P'_R) \) is c-planar.

We now show that if \( (G,P_B \cup P_R) \) is c-co-connected and \( (G,P'_R) \) is c-planar then \( (G,P_B \cup P_R) \) is c-planar: Let \( C \in P_B \cup P_R \). Being c-co-connected implies
Figure 3: An SPQR-tree $T$ rooted at a Q-node $r$ and its represented graph $G_r(r)$. The root edge of each skeleton is dashed. The correspondence of the remaining edges of the skeleton of a node $\mu$ and the edges around $\mu$ is as follows. If $\mu$ is a P-node, we assume that the edges around $\mu$ and the edges around the topmost pole of $\text{ske}l(\mu)$ are in the same clockwise order. If $\mu$ is an S-node then we assume that the edges in $\text{ske}l(\mu)$ from top to bottom correspond to the edges incident to $\mu$ from left to right. Finally, if $\mu$ is an R-node, we explicitly indicate the corresponding tree edges by little red arrows.

(e) A different embedding of $G_r(r)$ with the root edge on the outer face can be obtained by flipping the R-node and permuting the children of the P-node.
Figure 4: a) \( \mathcal{P}_R = \{ \{1\}, \{2,3,4,5,6,7\} \}, \mathcal{P}_B = \{ \{1,2,3,4\}, \{5,6,7\} \} \), i.e. clusters are separated by the curves. \((G, \mathcal{P}_B \cup \mathcal{P}_R)\) is c-co-connected and \(G\) is planar but \((G, \mathcal{P}_B \cup \mathcal{P}_R)\) is not c-planar. b) Clusters are enclosed by the curves. \((G, \mathcal{P}_B \cup \mathcal{P}_R)\) is c-connected and \((G, \mathcal{P}_B), (G, \mathcal{P}_R), (G, \mathcal{P}_I)\) are c-planar, but \((G, \mathcal{P}_B \cup \mathcal{P}_R)\) is not.

that \(V \setminus C\) is contained in a single face of \(G[C]\) in any planar drawing of \(G\). We call \(C\) bad if \(V \setminus C\) is contained in an inner face of \(G[C]\). We have to show that there is an embedding that has no bad clusters. Among all planar embeddings of \(G\) that are c-planar for \(\mathcal{P}_I\) choose one that has the minimum number of bad clusters. Assume there is a bad cluster \(C \in \mathcal{P}_B \cup \mathcal{P}_R\). We assume without loss of generality that \(C \in \mathcal{P}_B\) (squared blue vertices in the drawing). Let \(f\) be the face of \(G[C]\) containing \(V \setminus C\) (gray shaded area in the drawing).

If \(C\) is the union of some clusters in \(\mathcal{P}_R\), choose a face \(f_0\) of \(G\) inside \(f\) incident to a vertex of \(C\) as the outer face. Now \(C\) is not bad anymore and we do not create any new bad clusters. Thus, we decrease the number of bad clusters – contradicting that we started with an embedding with the minimum number of bad clusters.

Otherwise, let \(C' \in \mathcal{P}_R\) intersect \(C\) and \(V \setminus C\). Since \(G[C']\) is connected, \(E(C' \cap C, C' \setminus C)\) is not empty. There must even be an edge \(e \in E(C' \cap C, C' \setminus C)\) that is in the outer face of \(G[C']\): Otherwise \(G[C \cap C']\) would enclose \(C' \setminus C\), i.e., there is a cycle \(c\) in \(G[C \cap C']\) with a vertex in \(C' \setminus C\) in its interior. Thus \(f\) is contained in the region bounded by \(c\). However, \(c\) is contained in a connected component of \(C \cap C'\). This contradicts the fact that \((G, \mathcal{P}_I')\) is c-planar.

Let now \(f_0\) be a face of \(G\) incident to \(e\) in the outer face of \(G[C']\). Then \(f_0\) is incident to a vertex of the outer face of both a graph induced by a cluster in \(\mathcal{P}_R\) and a graph induced by cluster in \(\mathcal{P}_B\). Thus, \(f_0\) is not contained in any cluster. Choosing \(f_0\) as the outer face decreases the number of bad clusters by the same
Figure 5: A c-co-connected clustered graph \((G, C)\) with 3 partitions separated by the lines. In the respective coarsest common refinement, each connected component contains exactly one vertex. The graph is 3-connected, however, no choice of the outer face yields a c-planar embedding.

argument as above. This contradicts that we have chosen a planar embedding minimizing the number of bad clusters. □

Note that this method does not work for more than two partitions. Consider the example in Fig. 5.

Since c-planarity of c-connected hierarchically clustered graphs can be tested in linear time [11] and \((G, P_B \cup P_R)\) is c-connected and hierarchically clustered, it remains to show that \(P_I'\) can be constructed in linear time. Since connected components can be computed in linear time it suffices to show that the intersection partition \(P_I\) of the two partitions \(P_B = \{B_1, \ldots, B_{\ell_B}\}\) and \(P_R = \{R_1, \ldots, R_{\ell_R}\}\) of \(V\) can be computed in linear time. This might be common knowledge, but for the sake of completeness we give a quick description here:

We introduce the following data structure: For \(X \in \{B, R\}\), we use a vertex array with \(X[v] = i\) for \(v \in X_i\). We also initialize an array \(S[1, \ldots, \ell_R]\) of stacks, where \(S[i]\) will contain the vertices of \(R_i\), \(i = 1, \ldots, \ell_R\) in the order in which they appear in \(B_1, \ldots, B_{\ell_B}\). We fill the stacks as follows: For \(i = 1, \ldots, \ell_B\) and \(v \in B_i\), we push \(v\) to \(S[R[v]]\). Now, the sets in \(P_I\) can be obtained by going through the stacks and opening a new set whenever \(B[v]\) changes. This concludes the proof of the following theorem:

**Theorem 2** It can be tested in linear time whether a c-co-connected clustered graph is c-planar if the set of clusters is the union of two partitions of the vertex set.

Observe that if \((G, P_B \cup P_R)\) is only c-connected then \((G, P_B \cup P_R)\) does not have to be c-planar even if \((G, P_B)\), \((G, P_R)\), and \((G, P_I')\) are. See the example in Fig. 4b. Let \(G\) be the graph, and let \(P_R\) and \(P_B\), respectively, be the partition of the vertex set enclosed by the red dotted and blue dashed curves, respectively. Let \(P_I'\) be the connected intersection partition of \(P_R\) and \(P_B\). Then \((G, P_B)\), \((G, P_R)\), and \((G, P_I')\) are c-planar, but \((G, P_B \cup P_R)\) is not. This can be shown as follows:

The embedding in Fig. 4b is c-planar for \(P_B\) and \(P_I'\) – see also Fig. 6a and Fig. 6b. Fig. 6c shows an embedding that is c-planar for \(P_R\). Assume now that
Figure 6: Illustration why in Fig. 4b, $(G, P_B)$, $(G, P_R)$, and $(G, P'_I)$ are c-planar, but $(G, P_B \cup P_R)$ is not.

there would be an embedding that is c-planar for $(G, P_B \cup P_R)$. We use the vertex labeling indicated in Fig. 6. Due to cluster $\{1, 2, 3, 4\}$ the interior of the cycle $c_R = (1, 2, 3, 4)$ must be empty. Thus, vertex 5 must be drawn outside $c_R$. Due to the cluster $C = \{1, 3, 4, 5, 7\}$, vertex 2 and 6 must not be enclosed by the triangle $c_B = (1, 4, 5)$. It follows that the edges connecting 5 to $c_R$ must be drawn such that $c_B$ does not enclose $c_R$ and that 6 is outside $c_B$. Due to the edge $\{3, 7\}$, vertex 7 is not enclosed by $c_B$ either. Thus, except for the edge $e = \{1, 7\}$, the embedding is as indicated in Fig. 6c. But no matter how we would add $e$ in a planar embedding, we would either create a cycle in $G[C]$ enclosing vertex 2 or vertex 6, which are not in $C$.

4 C-Connected Clusterings on Blocks

We now present a polynomial-time method for testing c-planarity for a planar 2-connected graph $G$ and a set of (overlapping) c-connected clusters $C$. Throughout this section, let $T$ be the SPQR-tree of $G$ rooted at a Q-node $r$ representing the edge $e$ of $G$. We start with an informal description.

4.1 Informal Description

Recall that all planar embeddings of $G$ with $e$ on the outer face can be obtained by ordering the children of the P-nodes and by flipping the skeletons of the R-nodes. In order to find a suitable embedding, we will use a labeling scheme for each node $\nu$ in $T$ and its corresponding edge in the skeleton of its parent node to capture which parts of $\text{skel}(\nu)$ are contained in a given cluster.

Assume for a moment that the SPQR-tree does not contain R-nodes. In order to find a c-planar embedding, we have to permute the children of each P-node in such a way that for any cluster $C$ there is no $C$-cycle enclosing a vertex not in $C$. So consider a fixed P-node $\nu$ and a fixed cluster $C$. We call a $C$-cycle $c$ of $G_-(\nu)$ new if $c$ is not already contained in the graph represented by a child of $\nu$. We distinguish four cases for a child $\nu_i$ of $\nu$ (see Fig. 7).
Figure 7: In (a) node $\nu_3$ is double-border and all other nodes are outside. In (b) node $\nu_3$ is inside, $\nu_2$ and $\nu_4$ are border, and $\nu_1$ and $\nu_5$ are outside. The red area sketches the vertices within the cluster.

1. If $G^-_r (\nu_i)$ does not contain a $C$-path between its poles, it does not contribute to a new $C$-cycle, so we can put $\nu_i$ as one of the first or last children of $\nu$ and we call $\nu_i$ an outside node.

2. If $G^-_r (\nu_i)$ contains only vertices of $C$, then $G^-_r (\nu_i)$ can be contained in the interior of any $C$-cycle and we call $\nu_i$ an inside node.

3. If $G^-_r (\nu_i)$ contains a $C$-path between its poles, but also vertices not in $C$, we have to make sure that the latter are not enclosed by a $C$-cycle. We call $\nu_i$ border or double-border, depending on whether the vertices not in $C$ can all be put on the same side of the $C$-paths connecting the poles or not.

Now the children of $\nu$ must be permuted such that either there is one double-border node pre- and succeeded by arbitrary many outside nodes (see Fig. 7a), or there are arbitrary many consecutive inside nodes pre- and succeeded by at most one border node, respectively, and then arbitrary many outside nodes (see Fig. 7b).

Note that the method described in this section has some similarities with the algorithm described by Angelini et al. [1] for deciding whether a hierarchically clustered graph has a drawing with region-region crossings. They also use a labeling scheme for the nodes in the SPQR-tree. Their “full” corresponds to our “inside”, their “spined” corresponds to our “non-outside”. Observe, however, that their labels “side-spined” and “central-spined” depend on a given drawing while our labels “double-border” and “border” do not.

Depending on the labels, Angelini et al. permute the edges of the skeletons of each P-node and decide how to flip the children according to a 2-SAT formula. However, this requires that no two clusters determine different flips. This
property does no longer hold if clusters overlap as in our case. We discuss this problem in more detail in Sect. 4.3.4. Thus, in our approach we find the permutations of the children of multiple P-nodes simultaneously. In addition, we demonstrate how our method can be extended to the case where the underlying graph is not 2-connected (see Sect. 5).

In the next section, we describe our labeling scheme more formally. In Section 4.3, we then show how to use the consecutive ones property to check for all P-nodes and for all clusters simultaneously whether the required permutation can be found and whether the R-nodes can be flipped accordingly.

4.2 Formal Description and Characterization

In the following let $C \in \mathcal{C}$ be a cluster such that $G[C]$ is connected and let $C_{\text{ext}} \subseteq C$ be a subset of vertices that we want to be incident to the outer face of $G[C]$. (We will need $C_{\text{ext}}$ in Sect. 5 when extending the method to non-biconnected graphs. In that case, $C_{\text{ext}}$ will be a subset of the set of cut-vertices. In the case of biconnected graphs, $C_{\text{ext}}$ is empty.)

Let $\nu$ be a node in $T$ and let $s$ and $t$ be the poles of $G_r(\nu)$. We call a planar embedding of $G^{\prime}_r(\nu)$ appropriate if (a) the poles are on the outer face and (b) no $C$-cycle encloses a vertex of $(V \setminus C) \cup C_{\text{ext}}$.

Given a planar embedding of $G_r(\nu)$, let $f_1$ and $f_2$ be the two faces of $G_r(\nu)$ that are incident to the root edge. An outer $s$-$t$-path is the $s$-$t$-path in $G^{\prime}_r(\nu)$ that is incident to $f_1$ or $f_2$, respectively. Observe that there is one outer $s$-$t$-path if $\nu$ is a Q-node or an S-node that has only Q-nodes as children. Otherwise there are exactly two outer $s$-$t$-paths. E.g., in Fig. 9 the two outer $s$-$t$-paths of $G_{\{1,2\}}(S_2)$ are $\langle 2, 8, 7, 6, 4, 1 \rangle$ and $\langle 2, 6, 5, 1 \rangle$. We call a vertex of an outer $s$-$t$-path inner, if it is not incident to both faces, $f_1$ and $f_2$. I.e., the poles $s$ and $t$ are never inner. Further, if $\nu$ is an S-node then the poles of the children of $\nu$ are also not inner. E.g., vertex 5 is the only inner vertex of the outer $s$-$t$ path $\langle 2, 6, 5, 1 \rangle$ of $G_{\{1,2\}}(S_2)$ in Fig. 9.

Now, for a given cluster $C$, we label the node $\nu$ as

- **inside**, if all vertices of $G_r(\nu)$ are contained in $C$ and at most the poles $s$ and $t$ of $G_r(\nu)$ are in $C_{\text{ext}}$.
- **inappropriate**, if $G^{\prime}_r(\nu)$ has no appropriate embedding.
- **outside**, if it is not inappropriate and $G^{\prime}_r(\nu)$ contains no $C$-path between its poles.
- **border** if $\nu$ is neither inside nor outside and $G^{\prime}_r(\nu)$ has an appropriate embedding in which at most one of the outer $s$-$t$-paths contains inner vertices in $(V \setminus C) \cup C_{\text{ext}}$.
- **double-border** in all other cases.

See Fig. 8 for examples of the different labels in the case of a P-node and Fig. 9 for a more general example, both with $C_{\text{ext}} = \emptyset$. Let $\mu$ be a child of $\nu$ and let
$e_{\mu}$ be the edge of $\text{ske}l(\nu)$ corresponding to the edge $\{\nu, \mu\}$. Then $e_{\mu}$ gets the same label as $\mu$. We refer to nodes (and their corresponding edges) of any label except inside as non-inside and of any label except outside as non-outside.

Without computing any particular embedding and only by traversing the SPQR-tree $T$, we can decide for all nodes in linear time, whether they are inside, outside, border, double-border, or inappropriate for a given cluster $C$. To this end, we use the following terminology given an $R$-node $\nu$: Consider the unique embedding of $\text{ske}l^{-}(\nu)$ with the poles $s$ and $t$ on the outer face. A caging cycle is a simple cycle of non-outside edges in $\text{ske}l^{-}(\nu)$ that (a) contains double-border edges or (b) encloses vertices in $C_{\text{ext}}$ or non-inside edges. The following remark is a direct consequence of the definitions.

\textbf{Remark 1} An $R$-node $\nu$ is inappropriate if and only if one of its children is inappropriate or $\text{ske}l^{-}(\nu)$ contains a caging cycle.

\textbf{Lemma 1} Once the edges of the skeleton of an $R$-node $\nu$ without inappropriate children are labeled, it can be computed in time linear in the size of the skeleton, whether $\text{ske}l^{-}(\nu)$ contains a caging cycle.

\textbf{Proof:} First remove all outside edges of $\text{ske}l^{-}(\nu)$. Remove all but the edges incident to the outer face of the resulting graph and also remove all bridges. The remainder $S$ is a collection of edge-disjoint simple cycles each with empty interior (in $S$). If $S$ contains any double-border edge then there is a caging cycle. Otherwise consider all edges and vertices of $\text{ske}l^{-}(\nu)$ that were in the interior of one of the cycles of $S$. If among those there is a vertex in $C_{\text{ext}}$ or an edge labeled other than inside then there is a caging cycle. Otherwise there is no caging cycle. $\square$

\textbf{Lemma 2} For each cluster, the labels can be computed in time linear in the size of the input graph as summarized in Table 1.

\textbf{Proof:} Since the size of an SPQR-tree (including the sum of the sizes of the skeletons) is linear in the size of the represented graph, the linear run time follows directly from the construction and the previous remark. $\square$

An external path of a node $\nu$ is a path in $G$ between the poles of $G_{r}(\nu)$ that does not contain any other vertices of $G_{r}(\nu)$. We label the root edge of $\text{ske}l(\nu)$ inside for $C$ if $\nu$ has an external $C$-path and outside otherwise. We say that an external path $p$ of node $\nu$ is to the right (left) of $\nu$ with respect to the ordered pair $(s, t)$ of its poles if the cycle that is induced by $p$ in the graph that results from $G$ by contracting $G_{r}(\nu)$ into a single vertex – and removing multi-edges and loops – is oriented (counter-) clockwise assuming that $p$ was oriented from $t$ to $s$. Two external paths of $\nu$ are on the same side of $\nu$ if they are both to the right or both to the left of $\nu$ with respect to an arbitrary ordering of the poles of $\nu$. Otherwise, they are on different sides.

The following lemma characterizes the c-planar embeddings with respect to a single cluster.
Figure 8: Illustration of the labeling of a P-node $\nu$ depending on various given clusters $C$. The pictures show the graph $G_{r}^{\nu}(\nu)$. Blue encircled vertices are in $C$. We always assume $C_{\text{ext}} = \emptyset$. The node $\nu$ is labeled border in 8e, since $G_{r}^{\nu}(\nu)$ has an embedding (shown in 8f) that is c-planar for $\{C\}$ in which only one of the outer $s$-$t$-paths contains vertices in $C$. Similarly, the embedding in 8i shows why $\nu$ is labeled double-border in 8h.
Table 1: Computation of the labels of a node \( \nu \) with respect to a cluster \( C \), and a subset \( C_{\text{ext}} \subseteq C \), proceeding from the leaves to the root of the SPQR-tree. \( s \) and \( t \) are the poles. A node is inappropriate if and only if it is neither inside, nor outside, nor border, nor double-border.

<table>
<thead>
<tr>
<th>Node ( \nu ) is</th>
<th>inside</th>
<th>outside</th>
<th>border</th>
<th>double-border</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-node (leaf)</td>
<td>both vertices are in ( C ).</td>
<td>not both vertices are in ( C ).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P-node</td>
<td>all children are inside.</td>
<td>all children are outside.</td>
<td>( \nu ) has no double-border children and at most one border child.</td>
<td>( \nu ) has (a) two border children or (b) one double-border child and neither inside nor border children.</td>
</tr>
<tr>
<td>S-node</td>
<td>all children are inside.</td>
<td>( \nu ) has at least one outside child.</td>
<td>( \nu ) has no double-border children.</td>
<td>( \nu ) has at least one double-border child.</td>
</tr>
<tr>
<td>R-node</td>
<td>all children are inside.</td>
<td>skel(( \nu )) contains no caging cycle and no ( s )-( t )-path of non-outside edges.</td>
<td>skel(( \nu )) contains no caging cycle and exactly one outer ( s )-( t )-path contains non-inside edges or vertices from ( C_{\text{ext}} \setminus { s, t } ).</td>
<td>skel(( \nu )) contains no caging cycle.</td>
</tr>
<tr>
<td>Q-node (root)</td>
<td>the child of ( \nu ) is inside.</td>
<td>the child of ( \nu ) is outside and ( s \notin C ) or ( t \notin C ).</td>
<td>the child of ( \nu ) is (a) border or (b) outside and ( s, t \in C ).</td>
<td>the child of ( \nu ) is double-border and ( s \notin C ) or ( t \notin C ).</td>
</tr>
</tbody>
</table>
Figure 9: (a) A graph \( G \) with cluster \( C = \{1, 2, 3, 4, 6, 7\} \) containing the circled vertices and (b) the labeling of the nodes of its SPQR-tree: thick blue solid circled nodes are inside, dashed nodes border, and dotted nodes double-border. The root is inappropriate. The remaining nodes are outside. \( C_{\text{ext}} = \emptyset \).

Lemma 3 Let \( G = (V, E) \) be a 2-connected graph, let \( C \subseteq V \) be a cluster inducing a connected subgraph of \( G \), let \( C_{\text{ext}} \subseteq C \), let \( T \) be the SPQR-tree of \( G \), and let \( r \) be a Q-node of \( T \) representing an edge \( e \) of \( G \). A planar embedding of \( G \) with \( e \) on the outer face is c-planar for \( \{C\} \) with \( C_{\text{ext}} \) incident to the outer face of \( G[C] \), if and only if the following conditions are fulfilled for any non-inside node \( \nu \) of its SPQR-tree \( T \).

1. All external \( C \)-paths of \( \nu \) are embedded on the same side of \( G_r(\nu) \) – which is reflected by the side on which the root edge of skel(\( \nu \)) is embedded.

2. skel(\( \nu \)) contains no simple cycle of non-outside edges that encloses a non-inside edge or a vertex in \( C_{\text{ext}} \).

Proof: By definition, both conditions must be fulfilled for a c-planar embedding with \( C_{\text{ext}} \) on the outer face of the cluster. So assume now that both conditions are fulfilled. Let \( \nu \in (V \setminus C) \cup C_{\text{ext}} \), let \( e' \neq e \) be an edge incident to \( \nu \), and let \( \nu' \) be the Q-node representing \( e' \). Assume that \( G \) contains a \( C \)-cycle \( c' \) enclosing \( \nu \). By Condition 1, there is no node \( \nu \) on the \( \nu' \)-r-path such that \( e' \) can be decomposed into two external \( C \)-paths of \( \nu \). So, let \( \nu \) be the first node on the \( \nu' \)-r-path such that \( e' \) is contained in \( G^{-}_r(\nu) \) or can be composed by a path in \( G^{-}_r(\nu) \) and an external \( C \)-path of \( \nu \). Observe that \( c' \) induces a cycle \( c \) in skel(\( \nu \)) that contains only non-outside edges. Let \( \mu \) be the child of \( \nu \) on the \( \nu' \)-\( \nu \)-path. By the choice of \( \nu \) it follows that \( c \) does not contain the edge \( e_\mu \) of skel(\( \nu \)). Hence \( c \) encloses the edge \( e_\mu \). However, \( e_\mu \) was either not inside or \( \nu \in C_{\text{ext}} \) is an end vertex of \( e_\mu \) not in \( c \) – contradicting Condition 2.

In Sect. 4.3 we construct a set of binary matrices from an initial embedding of \( T \) that have the consecutive-ones property, if and only if there is a c-planar embedding for \( \mathcal{C} \) with the fixed root edge on the outer face. The ordering of the
columns of the matrices will correspond to the orderings of the children of the P-nodes and the flips of the R-nodes. The total size of the matrices will be in \( O(|V|^2|C|) \).

### 4.3 Modeling by Consecutive-Ones Property

For each possible root \( r \) of \( T \) that is not inappropriate for any \( C \in \mathcal{C} \), we start with a fixed embedding of \( T \) — including fixed flips of the R-nodes — and perform the following steps:

#### 4.3.1 Splitting \( T \)

We split \( T \) at each R-node (see Fig. 10), removing the edges from the R-node to its children from \( T \). Let \( T_r \) be the subtree containing \( r \). For each former non-leaf child \( \rho' \) of an R-node \( \nu \) we attach a new Q-node \( \rho \) to \( \rho' \). We root the subtree containing \( \rho' \) at \( \rho \) and denote it by \( T_\rho \). We label \( \rho \) inside for a cluster \( C \), if \( \rho' \) had an external \( C \)-path and outside otherwise. In the parent tree, we replace the R-node \( \nu \) by a special P-node \( \nu' \) with the same label and three Q-nodes \( \nu_1, \nu_2, \nu_3 \) in this order as children. If the R-node \( \nu \) was labeled border for a cluster \( C \), we label \( \nu_2 \) and exactly one among \( \nu_1 \) and \( \nu_3 \) as inside and the other as outside. More precisely, we label \( \nu_1 \) as outside if and only if the left outer path of skel\(^-\)(\( \nu \)) between its poles contains non-inside edges or vertices from \( C_{ext} \). If the R-node \( \nu \) was labeled double-border, we label \( \nu_1 \) and \( \nu_3 \) as border and \( \nu_2 \) as inside. If the R-node was labeled inside or outside, we label all three children as inside or outside, respectively. We thus end up with a forest containing only S-, P-, and Q-nodes.

#### 4.3.2 Initializing the Matrices

For each root \( \rho \) of one of the subtrees, we create a new binary matrix \( M_\rho \). A node in \( T_\rho \) is a **lowest-P-child**, if it is the child of a P-node and has no other P-nodes in its subtree. The embedding of \( T_\rho \) induces an ordering of the lowest-P-children from left to right. We initialize \( M_\rho \) with a column for each lowest-P-child in accordance with the ordering. For a node \( \nu \) of \( T_\rho \), we use \( c(\nu) \) to refer to the set

![Figure 10: The tree \( T \) is split at every R node, resulting in a forest containing only S-, P-, and Q-nodes.](image)
If an S-node's subtree is filled with 1s. See Fig. 12a.

For each cluster $C$, one of the two external columns will represent the side of possible external $C$-paths of the child of $\rho$ and will be denoted by $c_C(\rho)$.

We then create a row for each non-leaf node $\nu$ distinct from a lowest-P-child, adding 1s in the columns in $c(\nu)$ and 0s in all other columns. This ensures that in every permutation of the columns of $M_\rho$ for which the 1s are consecutive in all rows, the columns of the lowest-P-children of each node remain adjacent, allowing a reconstruction of an embedding of $T_\rho$ from the ordering of the columns in $M_\rho$. See Fig. 11. If $\rho \neq r$ we add two rows having all 1s except for one 0 in the first or last external column, respectively.

In order to fill the matrix $M_\rho$, we traverse the tree $T_\rho$ with a post-order DFS. For each cluster $C \in \mathcal{C}$ and each examined node we add up to three rows to $M_\rho$.

For each node $\nu$ and each cluster $C$ a set $r(\nu, C)$ of relevant rows. For each lowest-P-child $\nu$, we set $r(\nu, C) = \emptyset$. The block $B(\nu, C)$ is the submatrix of $M_\rho$ with entries in rows $r(\nu, C)$ and columns $c(\nu)$. When we create rows in $M_\rho$, the default entries are 0 and we explicitly mention when we set the entries to 1.

### 4.3.3 Handling P-nodes

For a P-node $\nu$ with children $\nu_1, \ldots, \nu_k$ we initialize $r(\nu, C)$ as $r(\nu_1, C) \cup \ldots \cup r(\nu_k, C)$. Since $\nu$ is not inappropriate, we have one among the following two cases: a) $\nu$ has only outside children except for possibly one double-border child. Then the children of $\nu$ can be permuted arbitrarily. b) the children of $\nu$ must be permuted such that all inside children are consecutive pre- and succeeded by at most one border child and arbitrary many outside children.

Thus, if $\nu$ is not outside and has no double-border child, we add up to 3 constraint-rows $r_0(\nu, C)$, $r_1(\nu, C)$, and $r_2(\nu, C)$ to $r(\nu, C)$. If $\nu$ has inside children, we add $r_0(\nu, C)$ with 1s in all columns in $c(\nu_i)$ where $\nu_i$ is an inside child of $\nu$. This ensures that all inside children are placed in consecutive order. If $\nu$ has a child $\mu$ that is a border node, we add $r_1(\nu, C)$ with 1s in all columns in $c(\mu)$ and again with 1s in all columns in $c(\nu_i)$ where $\nu_i$ is an inside child of $\nu$. We do the same for a potential second border child in a third row $r_2(\nu, C)$. This ensures that the border children are placed next to the inside children, with at most one border child on each side. Finally, let $\mu$ be a child of $\nu$ let $i \in r(\mu, C)$ and let $j \in c(\nu) \setminus c(\mu)$. Then we set the entry in row $i$ and column $j$ to 1, if one or more of the rows in $r(\nu, C) \setminus r(\mu, C)$ contain a 1 in the same column. This is to make sure that the columns between the P-node and potential ones induced by external paths is filled with 1s. See Fig. 12a

### 4.3.4 Handling S-nodes

If an S-node $\nu$ with children $\nu_1, \ldots, \nu_k$ is outside then $r(\nu, C) = \emptyset$, otherwise $r(\nu, C) = r(\nu_1, C) \cup \ldots \cup r(\nu_k, C).$
Assume now that \( \nu \) is not outside and has an external \( C \)-path. Observe that in this case \( \nu \) cannot be double-border. Otherwise \( r \) would be inappropriate for \( C \). If \( \nu \) has two or more P-nodes as children, we have to make sure that the 1s in each P-node and the 1s in the external path can be made consecutive via additional 1s.

More precisely, let \( \nu_1, \nu_2, \ldots, \nu_k \) be the children of \( \nu \) that are P-nodes. The upper half for a cluster \( C \) are all entries in rows \( r(\nu_i, C) \), \( i = 1, \ldots, k \) and columns \( c(\nu_j) \), \( j = i + 1, \ldots, k \) while the lower half are all entries in rows \( r(\nu_i, C) \), \( i = 1, \ldots, k \) and columns \( c(\nu_j) \), \( j = 1, \ldots, i - 1 \). We fill both, the upper and the lower half with 1s if \( \nu \) is inside and we fill either the upper or the lower half with 1s if \( \nu \) is border. See Fig. 12b.
Recall that if $\nu$ is not inside then the external $C$-paths must all be on the same side of $G_r(\nu)$ in a c-planar drawing of $G$. However, external $C_1$- and $C_2$-paths could be on different sides for distinct clusters $C_1$ and $C_2$. Hence, we cannot just always fill the upper half with 1s. To this end, we will define same and different constraints on the clusters that are critical for $\nu$, i.e., on the clusters $C$ such that $\nu$ is border with respect to $C$ and has an external $C$-path.

Let $C_1$ and $C_2$ be two clusters that are critical for $\nu$. Then we say that there is a same (different) constraint between $C_1$ and $C_2$ if the external $C_1$-paths and the external $C_2$-paths must be on the same side of $G_r(\nu)$ in any c-planar embedding of $G$. The following remark implies that there is either a same or a different constraint between any two clusters that are critical for $\nu$ and how to decide whether there is a same or a different constraint.

**Remark 2** Let $C_1$ and $C_2$ be two clusters that are critical for $\nu$. If $\nu$ has an external $C_1$-path that is also an external $C_2$-path then there is a same constraint between $C_1$ and $C_2$. Otherwise, there is a different constraint between $C_1$ and $C_2$.

**Proof:** Assume that there is an external $C_1$-path $p_1$ and an external $C_2$-path $p_2$ of $\nu$ that are on the same side of $G_r(\nu)$ in a c-planar embedding of $G$.

Since $\nu$ is border, there is a $C_i$-path $p_i^r$, $i = 1, 2$ in $G_r(\nu)$. Consider the cycles $c_i$, $i = 1, 2$ composed by $p_i^r$ and $p_i$. By c-planarity, each portion of $p_1$ that is inside $c_2$ must be in $G[C_2]$ and vice versa. Since $p_1$ and $p_2$ are on the same side of $G_r(\nu)$, there is an external path of $\nu$ that contains only edges of $p_1$ inside $c_2$, edges of $p_2$ inside $c_1$ and common edges of $p_1$ and $p_2$, i.e., only edges in $G[C_1 \cap C_2]$. \(\square\)

Observe that for every S-node $\nu$ and every pair $C_1$, $C_2$ of clusters, we can decide in $O(|V|)$ time whether $\nu$ has an external $(C_1 \cap C_2)$-path, and thus, whether there is a same-constraint between $C_1$ and $C_2$. 

![Figure 12: (a) The block $B(\nu,C)$ for a P-node $\nu$ with an inside child $\nu_2$, an outside child $\nu_4$ and two border children $\nu_1$ and $\nu_3$. (b) The block $B(\nu,C)$ for an S-node $\nu$ with four children. If $\nu$ is not outer and has an external $C$-path then the upper half, the lower half, or both are filled with 1s.](image)
Fix now an arbitrary cluster $C$ that is critical for $\nu$ and assign $C$ the upper half. Assign to any other cluster $C'$ that is critical for $\nu$ the upper half if there is a same constraint between $C$ and $C'$ and the lower half otherwise.

4.3.5 External Columns

If $\rho = r$ (i.e. the subtree has no parent) let $\nu$ be the unique child of $r$ and let $e$ be the edge represented by the Q-node $r$. Then the external column is 1 for each row in $r(\nu, C)$ if the cluster $C$ contains both end vertices of $e$.

If $\rho \neq r$ then the unique child $\rho'$ of $\rho$ was the child of an R-node $\nu$. Consider the fixed embedding of skel$^-(\nu)$ with its poles $s$ and $t$ on the external face. Let $C$ be a cluster for which $\rho'$ is a border node and has an external $C$-path. We have to make sure that the parts of $G_r(\rho')$ that are not in $C \setminus C_{ext}$ are embedded such that they are not enclosed by a $C$-cycle in $G$ that is composed of an external $C$-path of $\rho'$ and a $C$-path in $G_r(\rho')$ between its poles $s'$ and $t'$.

Consider first that skel$^-(\nu)$ contains a simple cycle $c$ containing $e_{\rho'}$ and consisting only of non-outside edges. If $c$ is (counter-)clockwise oriented when orienting $e_{\rho'}$ from $s'$ to $t'$, then we set $c_C(\rho)$ to be the (left) right external column.

Otherwise all external $C$-paths of $\rho'$ must contain an external $C$-path of $\nu$. Thus, $\nu$ is not double-border. Moreover, the set of vertices of skel$^-(\nu)$ that can be reached from $s$ using only non-outside edges and not $e_{\rho'}$ induces an $s$-$t$-cut of skel$^-(\nu)$ that contains $e_{\rho'}$ and no other non-outside edges. It follows that $e_{\rho'}$ is on the left (right) outer $s$-$t$-path and all external $C$-paths of $\nu$ are to the left (right) of $\nu$ with respect to $(s,t)$ in any c-planar embedding. Hence, if $e$ is on the left (right) outer $s$-$t$-path then we set $c_C(\rho)$ to be the left (right) external column. In both cases we set the entry in column $c_C(\rho)$ to 1 for each row in $r(\rho', C)$.

Clearly the number of columns is linear in the number of Q-nodes and R-nodes and thus linear in $|V|$ for planar graphs. During initialization, we add $O(|V|)$ rows, namely one for each inner node. For a cluster $C$ and a P-node $\nu$, we add up to three rows but only if both poles are in $C$. Observe that at least one of the poles of a P-node $\nu$ is not a pole of another P-node $\nu'$ on the path from $\nu$ to the root. Hence, the number of rows added for P-nodes is bounded by $3 \cdot \sum_{C \in C} |C| \in O(|V| \cdot |C|)$.

Applying the next theorem with $C_{ext} = \emptyset$ yields a characterization of c-connected overlapping clustered graphs with underlying 2-connected graphs.

Theorem 3 A c-connected overlapping clustered graph $(G, C)$ with an underlying planar 2-connected graph $G$ and sets $C_{ext} \subseteq C$, $C \in C$ has a c-planar embedding in which $C_{ext}$ is incident to the outer face of $G[C]$ for any $C \in C$ if and only if the root of the SPQR-tree of $G$ can be chosen such that it is not inappropriate for $C \in C$ and all matrices $M_{\rho}$ fulfill the consecutive-ones property.

We prove the theorem in the following two subsections.
4.4 Proof of Sufficiency

Let the SPQR-tree $T$ of $G$ be rooted at the Q-node $r$, and let $e$ be the edge represented by $r$. Assume that $r$ is not inappropriate for any cluster and that the columns of all matrices $M_\rho$ are permuted such that in each row the 1s are consecutive. We have to show that $(G, C)$ is c-planar.

We may assume without loss of generality that the external columns were not permuted. (Otherwise reverse the order of the columns.) Starting from $\rho = r$, we traverse $T$ and do the following at a non-leaf node $\nu$:

If $\nu$ is a P-node, we permute the children $\nu_1, \ldots, \nu_k$ of $\nu$ according to the ordering of $c(\nu_1), \ldots, c(\nu_k)$ in the permuted matrix $M_\rho$.

If $\nu$ is an R-node, we fixed an embedding of $G_r(\nu)$ and replaced $\nu$ with a P-node and three incident Q-nodes $\nu_1, \nu_2, \nu_3$ in this order. If $\nu$ was labeled inside or outside for all clusters then we maintain the fixed flip of $G_r(\nu)$. Otherwise the labeling was such that $c(\nu_2)$ will remain between $c(\nu_1)$ and $c(\nu_3)$. We maintain the fixed embedding of $G_r(\nu)$ if $c(\nu_1)$ remains before $c(\nu_3)$ after the permutation and flip $G_r(\nu)$ otherwise. If we flip $G_r(\nu)$, we also reverse all matrices for all non-leaf nodes in the subtree rooted at $\nu$ that are children of an R-node. Finally, we embed $e$ to the right of $G_r(-r)$ if the external column of $M_r$ is on the right hand side of $M_r$ and to the left otherwise.

**Lemma 4** The thus constructed embedding of $(G, C)$ is c-planar.

**Proof:** Let $C \in C$ and let $\nu_1$ be a non-inside node of $T$. We show by induction on the length of the $\nu_1$-r-path that all external $C$-paths of $\nu_1$ are on the same side and that no non-inside edge and no vertex in $C_{\text{ext}}$ is enclosed by a simple cycle of non-outside edges in ${\text{ske}(\nu_1)}$ – provided that the root edge of $\text{ske}(\nu)$ is embedded on the same side as the external $C$-paths of $\nu_1$. I.e., we have to consider the following three cases of pairs of $C$-paths between the poles of $\nu_1$ (see Fig. 13).

(a) $\nu_1$ has two external $C$-paths $p_1$ and $p_2$. 

Figure 13: The three forbidden cases in the proof of Lemma 4.
that the cycle composed by

\[ \nu \]

\[ G \]

If \[ C \]

\[ \nu \]

the R-node and we redefine \[ \nu \]

Let \[ p_j, j = 1, 2 \] be C-paths in \[ G^- (\mu_j) \] between its poles.

\( \text{skel}^- (\nu_1) \) contains a cycle \( c \) of non-outside edges. Observe that if \( c \) enclosed a non-inside edge or a vertex in \( C_{\text{ext}} \) then \( \nu_1 \) could not be an R-node; otherwise \( c \) would be a caging cycle and thus, \( \nu_1 \) and all its ancestors, including the root, would be inappropriate. Thus, in this case we may assume that \( \nu_1 \) is a P-node with two non-outside children \( \mu_1, \mu_2 \). Let \( p_j, j = 1, 2 \) be C-paths in \( G^- (\mu_j) \) between its poles.

\( \text{skel}(\nu_1) \) contains a cycle \( c \) of non-outside edges such that the root edge is contained in \( c \). Then there are the following two C-paths between the poles of \( \nu_1 \): An external C-path \( p_2 \) of \( \nu_1 \) and a path \( p_1 \) within \( G^- (\nu_1) \) that goes exactly through the graphs represented by the children of \( \nu_1 \) that correspond to the edges of \( c \) other than the root edge.

Let now \( e_0 \) be a non-inside edge of \( \text{skel}(\nu_1) \) or let \( v_0 \in C_{\text{ext}} \) be a vertex of \( \text{skel}(\nu_1) \) other than the poles and let \( e_0 \) be an edge of \( \text{skel}(\nu_1) \) incident to \( v_0 \). Let \( v_0 \) be the child of \( \nu_1 \) corresponding to \( e_0 \). In all three cases, we have to show that the cycle composed by \( p_1 \) and \( p_2 \) does not enclose \( G^- (\nu_0) \).

We will introduce a notation such that we can handle all three cases in one step. First, we will split the path \( p_j, j = 1, 2 \) into three segments, some of which might be empty. Figure [1] gives an illustration of the most general case.

Let \( \nu_1, \ldots, \nu_t = r \) be the \( \nu_1 \)-r-path. Let \( j \in \{1, 2\} \). If \( p_j \) in \( G^- (\nu_1) \) let \( i_j = 1 \). Otherwise let \( 2 \leq i_j \leq \ell \) be minimum such that \( \nu_{i_j} \) is an R-node or \( G^- (\nu_{i_j}) \) contains \( p_j \). We may assume that \( i_1 \leq i_2 \). If \( \nu_{i_2} \) is an R-node, we actually redefine \( \nu_{i_2} \) to be the root \( \rho \) of the tree containing \( \nu_{i_2-1} \): we replace \( p_j \) by the respective path in \( G_p (\nu_{i_2-1}) \) through the root edge \( e_\rho \) of \( \text{skel}(\nu_{i_2-1}) \). If \( \nu_1 \) was an R-node we redefine \( \nu_1 \) to be the special P-node with which we replaced the R-node and we redefine \( v_0 \) to be one of the artificial non-inside Q-nodes we appended to \( \nu_1 \).

Observe that \( \nu_{i_2} \) is either \( \rho \) or a P-node and \( p_j \) is composed by two C-paths \( p_1^j \) and \( p_2^j \) connecting the poles of \( G_r (\nu_1) \) with the poles of \( G_r (\nu_{i_2}) \) and a middle C-path \( p_1^j \). \( p_1^j \) and \( p_2^j \) are empty if \( i_j = 1 \). \( p_1^j \) consists of the edge \( e_\rho \) if \( \nu_{i_2} = \rho \).

If \( \nu_{i_2} \) is a P-node then it has a non-outside child \( \mu_j \neq \nu_{i_2-1} \) such that \( p_1^j \) is a path in \( G^- (\mu_j) \) between its poles.

We distinguish some cases. (1) If \( \nu_1 = \rho \) or if \( \nu_1 = \nu_{i_2} \neq \rho \) and \( \mu_1 = \mu_2 \) then \( p_1 \) and \( p_2 \) are trivially on the same side of \( G_r (\nu_1) \). (2) Assume that \( \nu_{i_1} = \nu_{i_2} \neq \rho \) and \( \mu_1 \neq \mu_2 \). Since the 1s are consecutive in the rows inserted for \( \nu_{i_2} \), the two non-outside children \( \mu_1 \) and \( \mu_2 \) must be on the same side of the non-inside child \( \nu_{i_2-1} \). (3) Otherwise, observe that the C-paths \( p_1^2 \) and \( p_2^2 \) connecting the poles of \( G_r (\nu_1) \) with the poles of \( G_r (\nu_{i_2}) \) must contain the poles of \( G_r (\nu_k) \), \( k = 1, \ldots, i_2 \). (See Fig. [1]b) This implies especially that for each \( k = i_1, \ldots, i_2 \) the graphs \( G^- (\nu_k) \) contain a C-path connecting their poles: such a C-path can be composed by \( p_1^1 \) and portions of \( p_1^2 \) and \( p_2^2 \). Hence, \( \nu_k, k = i_1, \ldots, i_2 \) is not outside. Further a subpath of \( p_2 \) is an external C-path of \( \nu_{i_2} \). Hence, \( \nu_1 \) cannot be double-border, since otherwise the root would be inappropriate for \( C \).

Since \( \nu_{i_2-1} \) is non-inside and \( \nu_{i_2} \) is border or inside there is a row \( \kappa \) inserted for \( \nu_{i_2} \) that contains only 0s in \( c(\nu_{i_2-1}) \) and only 1s in \( c(\mu_1) \). Further, when we
handled $\nu_{i_2}$, we added 1s in the row $\kappa$ and the external column (if $\nu_{i_2} = r$) or the columns $c(\mu_2)$ (otherwise). Hence, since the 1s must be consecutive in $\kappa$, it follows that $c(\nu_{i_1} - 1)$ cannot be between $c(\mu_1)$ and the external column $c_C(\rho)$ or $c(\mu_2)$, respectively. Hence, $p_1$ and $p_2$ must be on the same side of $G_r(\nu_0)$.

Now, if $\nu_{i_2} \neq \rho$ we are done. Otherwise let $\nu$ be the parent R-node of $\nu_{i_2 - 1}$ in $T$. By induction, we already know that all external $C$-paths of $\nu$ are on the same side and that $e_{\nu_{i_1} - 1}$ is not enclosed by a simple cycle of non-outside edges in skel($\nu$). Hence, the external $C$-paths of $\nu_{i_2 - 1}$ are all on the same side and by construction this is represented by the external column $c_C(\rho)$. \hfill \Box

4.5 Proof of Necessity

Assume now that a c-planar embedding with $e$ on the outer face is given in which $C_{\text{ext}}$ is on the outer face of $G[C]$ for every cluster $C$. This yields a permutation of the children of the $P$-nodes of $T$ and flips of the R-nodes. Permute the columns of the matrices accordingly. Let $\rho$ be the root of a split off tree $T_\rho$ and let $\rho'$ be the only child of $\rho$ in $T_\rho$. The external columns of $M_\rho$ are exchanged if on the $\rho'$-$r$-path there are an odd number of R-nodes that are flipped.

Lemma 5 The thus constructed permutation of the columns is a consecutive-ones ordering up to a possible re-permutation due to the assignment of upper and lower halves for S-nodes.
Proof: Recall that we have inserted up to three rows for each P-node and each cluster and no other rows into the matrices (except for the initialization). Let \( \nu \) be a P-node in a subtree \( T_\rho \) and let \( C \) be a cluster such that we have created a row \( \kappa \) for \( \nu \) and \( C \) in \( M_\rho \). Then \( \nu \) has no double-border child. Due to c-planarity and the condition on all \( C_{ext} \), the children of \( \nu \) must be permuted such that all inside children are consecutive pre- and succeeded by at most one border child and arbitrary many outside children. It follows that the 1s in columns \( c(\nu) \) must be consecutive.

Let \( \nu = \nu_1, \ldots, \nu_k = \rho \) be the path from \( \nu \) to the root of \( T_\rho \), and let \( 1 < k \leq \ell \) be maximum such that \( \nu_1, \ldots, \nu_k \) are not outside. If \( \nu_1 \) was not a special P-node substituting an R-node then \( \nu_1 \) is a P-node if \( i \) is odd and an S-node if \( i \) is even (otherwise it might be vice versa, but the situation is similar). \( \nu_k \) is a P-node if \( \nu_k \neq \rho \). Also observe that \( c(\nu_{i-1}) \subseteq c(\nu_i) \), \( i = 2, \ldots, \ell \) and that for each \( i = 1, \ldots, \ell \) the columns in \( c(\nu_i) \) are consecutive in the permuted matrix. If \( k < \ell \), we have set \( \tau(\nu_{k+1}, C) = \emptyset \). Hence, the entries in row \( \kappa \) are 0 in all columns in \( c(\nu_{\ell-1}) \setminus c(\nu_k) \).

We consider first a P-node \( \nu_i \), \( i = 3, \ldots, k \) odd. Since \( \nu_{i-1} \) is not outside it follows that no child of \( \nu_i \) other than \( \nu_{i-1} \) can be double-border. Hence, for each non-outside child \( \mu \neq \nu_{i-1} \) of \( \nu_i \) there are 1s in row \( \kappa \) and all columns in \( c(\mu) \).

Observe that due to c-planarity the non-outside children of \( \nu_i \) are consecutive. Moreover, if there are both, non-outside children of \( \nu_i \) to the right and the left of \( \nu_{i-1} \) then \( \nu_{i-1} \) is inside and, thus, all columns in \( c(\nu_1) \) as well as \( c(\nu_j) \setminus c(\nu_{j-1}) \) have entry 1 in row \( \kappa \) for all \( 3 < j < i \) odd.

If \( \nu_k = \rho \), let \( m = k = \ell \) and assume that the external \( C \)-paths of \( \nu_{\ell-1} \) are all to the right (left) of \( \nu_{\ell-1} \), i.e., the column \( c_C(\rho) \) is the right (left) external column. If \( \nu_k \neq \rho \), let \( m \leq k \) be maximum such that \( \nu_m \) is a P-node that has a non-outside child other than \( \nu_{m-1} \) (If no such P-node exists then all entries in row \( \kappa \) other than in the columns \( c(\nu) \) are zero and thus all 1s are consecutive.) Assume that \( \nu_m \) has a non-outside child \( \mu \) to the right (left) of \( \nu_{m-1} \). Assume now that there is a \( 1 \leq j < m \) odd such that the P-node \( \nu_j \) has a child \( \mu' \) to the right (left) of \( \nu_{j-1} \). I.e., the columns \( c(\mu') \) are between the columns \( c(\nu) \) and \( c(\mu) \). If \( \mu' \) were not inside then \( G_r(\mu') \) would contain a vertex in \( V \setminus (C \setminus C_{ext}) \) that would be enclosed by a \( C \)-cycle composed by the following four paths: (1) A \( C \)-path in \( G_r(\nu_{j-1}) \) between its poles, (2+3) two \( C \)-paths connecting the poles of \( G_r(\nu_{m-1}) \) with the poles of \( G_r(\nu_j) \), and (4) either an external \( C \)-path of \( \nu_{\ell-1} \), if \( \nu_m = \rho \) or a \( C \)-path in \( G_r(\mu) \) between its poles, if \( \nu_m \) is a P-node. Hence, the entries in \( c(\mu') \) are all 1.
Consider now an S-node $\nu_i$, $i = 2, \ldots, k$ even, that has an external $C$-path. By the choice of $k$, $\nu_i$ is not outside. Since the root is not inappropriate, $\nu_i$ is not double-border. Thus, we've set the entries in row $\kappa$ and columns $c(\nu_i) \setminus c(\nu_{i-1})$ to 1 if $\nu_i$ is inside. Otherwise, we set the entries in $c(\nu_i) \setminus c(\nu_{i-1})$ that are to one side of $c(\nu_{i-1})$ to 1. Observe that an S-node $\nu_i$ has an external path if and only if $\ell = k$ or $i < m$.

Hence, row $\kappa$ looks as follows: Assume without loss of generality that $c(\nu_m) \setminus c(\nu_{m-1})$ contains a 1 to the right of $c(\nu_{m-1})$. Then the entries in $c(\nu)$ are ordered such that all 0s (if any) are to the left and all 1s are to the right. Moreover, if $\nu$ is inside let $1 \leq b \leq k$ be maximal such that $\nu_b$ is inside. Then all entries in columns $c(\nu_b)$ are 1. Otherwise let $b = 1$.

For $i = b+1, \ldots, m-2$ odd all entries in $c(\nu_i) \setminus c(\nu_{i-1})$ that are on the right side of $c(\nu_{i-1})$ are 1. For $i = b+1, \ldots, m-1$ even, all entries in $c(\nu_i) \setminus c(\nu_{i-1})$ on one side of $c(\nu_{i-1})$ are 1 – however, for some $i$ that could be the right-hand side and for others the left-hand side. Finally, the entries in $c(\nu_m) \setminus c(\nu_{m-1})$ to the right of $c(\nu_{m-1})$ are ordered such that the 1s are to the left and the 0s (if any) are to the right. See Fig. 15.

Hence, the 1s in row $\kappa$ are consecutive up to maybe the wrong choice of the side for the 1s inserted for border S-nodes. Observe, however, on one hand that we could remove now the 1s from the wrong side and insert them on the right side and would thus obtain the 1s consecutive. We could obtain that for one cluster also by permuting the columns for the children of the S-node accordingly. On the other hand the assignment to sides was forced by the same and different constraints – up to the choice for one cluster. Hence, if we do the permuting that works for one cluster it will create the feasible assignment we would obtain if we had assigned the sides now that we know were the external paths are embedded. □

5 C-Connected Clusterings on General Graphs

We now show how to extend the method from the last section to work for an arbitrary planar graph. Let $(G, \mathcal{C})$ be a $c$-connected overlapping clustered graph with underlying planar graph $G$. If $G$ is not connected, we can test each
Figure 16: The hierarchy in the BC-tree given by the choice of the root. The green vertex is the parent cut vertex of $B$ while the red vertex is a child cut vertex of $B$. Note that the child blocks must be embedded with their parent cut vertices on the outer face.

connected component separately, since the $c$-connectivity limits each cluster to a single component.

It remains to show the case where $G$ is connected but not 2-connected and can thus be represented by a BC-tree. We consider the BC-tree of $G$ rooted at a block $H_r$. The interpretation of this choice is that $H_r$ should contain an edge incident to the outer face of $G$ in a planar drawing. Let $H$ be a block of $G$. If $H \neq H_r$ then the parent cut vertex of $H$ is the cut vertex of $H$ on the path from $H$ to $H_r$. $H$ is a child block of its parent cut vertex. All other cut vertices of $H$ are called child cut vertices of $H$. All cut vertices of $H_r$ are child cut vertices of $H_r$.

Consider the SPQR-tree $T$ of $H$. If $H = H_r$, any root of $T$ is suitable. Otherwise the parent cut vertex of $H$ must be on the outer face of $H$. Thus, a root of the SPQR-tree $T$ is suitable if it corresponds to an edge incident to the parent cut vertex of $H$. See Fig. 16.

Let $H_1, \ldots, H_k$ be the child blocks of a cut vertex $v$ and let $V_i$, $i = 1, \ldots, k$ be the set of vertices in the connected components of $G - v$ containing $H_i$. We call a cluster $C$ relevant for a child block $H_i$, if $v \in C$ and $V_i \not\subseteq C$. Let $C_{ext}$ be the set of child cut vertices $v$ of $H$ such that $C$ is relevant for a child block of $v$.

We use the algorithm for 2-connected graphs, restricting the roots for the SPQR-trees to be suitable, to test whether there is some $c$-planar embedding for each block $H$ with the parent cut vertex on the outer face of $H$ and $C_{ext}$ on the outer face of $H[C]$. For each child cut vertex $v$ of a block $H$ and for each child block $H_i$ of $v$, we test whether there is a free face, i.e., a face $f$ of $H$ incident to $v$ such that the boundary of $f$ contains a vertex not in $C$ for any cluster $C$ that is relevant for $H_i$. If so, the $c$-planar embeddings of the blocks can be combined into a $c$-planar embedding of the whole graph. In the following, we show that otherwise there is no $c$-planar drawing for the whole graph with the given choices of the root of the BC-tree and the roots of the SPQR-trees.

Given a $c$-planar embedding, a face $f$ is free with respect to a subset $C' \subseteq C$ of clusters if $f$ is not enclosed by a $C$-cycle for any $C \in C'$. Otherwise, $f$ is
Remark 3 A face is covered by $C'$ if and only if its boundary is a $C$-cycle.

A vertex $v$ is free with respect to a subset $C' \subseteq C$ of clusters if one of its incident faces is free with respect to $C'$ and covered by $C'$ otherwise. Our goal is to prove the following lemma.

Lemma 6 Let $H$ be a block, $v$ a vertex in $H$, and $C' \subseteq C$. If there is a c-planar embedding of $H$, in which $v$ is free with respect to $C'$, then $v$ is free with respect to $C'$ in any c-planar embedding of $H$ in which $v$ is free with respect to $C$ for all $C \in C'$.

Given a vertex $v$ in a block $H$, we call two incident edges $e_1$ and $e_2$ of $v$ equivalent with respect to a set $C'$ of clusters, if they are in the same block of $H[\bigcap_{C \in C'} C]$, i.e. if there is a simple cycle in $H$ that is a $C$-cycle for any $C \in C'$ and contains both, $e_1$ and $e_2$. A $C'$-equivalence class around $v$ is a maximal set of edges incident to $v$ that are pairwise equivalent with respect to $C'$.

Lemma 7 Let $v$ be a vertex of a block $H$ and let $C' \subseteq C$. Then a $C'$-equivalence class around $v$ is a consecutive set of edges in the cyclic order around $v$ in any c-planar embedding of $H$.

Proof: Let $C'$ be a set of clusters, $v$ a vertex in block $H$ and let $e_i = \{v, v_i\}$, $i = 1, 2$ be two edges incident to $v$ that are contained in a simple cycle $c$ in $H[\bigcap_{C \in C'} C]$. See Fig. 17a. Then all vertices that are enclosed by $c$ are in $\bigcap_{C \in C'} C$. Let $e' = \{v, v'\}$ be an edge enclosed by $c$. Let $i \in \{1, 2\}$. Since $H$ is 2-connected, there must be a $v'$-$v_i$ path $p_i$ in $H$ not containing $v$. Let $v'_i$ be the first vertex of $p_i$ on $c$. Let $c'_i$ be the cycle formed by the $v'_i$-$v'_i$-subpath of $p_i$, the $v'_i$-$v$-$v_i$-subpath of $c$ containing $e_i$ and the edge $e'$. Then $c'_i$ is a simple cycle in $H[\bigcap_{C \in C'} C]$ containing $e'$ and $e_i$. \hfill \Box

Let $v$ be a vertex that is free with respect to any $C \in C'$. Given a c-planar embedding of a block, a $C'$-interval around a vertex $v$ is a maximal sequence of consecutive edges around $v$ that are (a) equivalent with respect to $C'$ and such that (b) the face between any two consecutive edges is covered by $C$ for all $C \in C'$. Note that there is a one-to-one correspondence between the $C'$-equivalence classes and the $C'$-intervals around $v$: the condition that $v$ is free with respect to any cluster in $C'$ guarantees that the $C'$-intervals have a well defined start and end point. Also note that there might be several distinct $C'$-intervals around $v$ – even if $C'$ contains only one cluster.

Proof: (of Lemma 6) Assume that there is a c-planar embedding of $H$ in which $v$ is free with respect to $C$ for all $C \in C'$ but $v$ is not free with respect to $C'$. Consider the cyclic order $e_1, \ldots, e_\ell$ of the edges around $v$. Since $v$ is not covered by any $C \in C'$, the $C$-intervals around $v$ are well defined. Among all $C$-intervals for all $C \in C'$, let $I$ be a minimal set of intervals such that all faces around $v$ are covered by at least one interval in $I$. Let $I_i = \langle e_{s_i}, \ldots, e_{t_i} \rangle$, $i = 1, \ldots, \kappa$
be the intervals in $\mathcal{I}$ in cyclic order around $v$. See Fig. 17b. We assume that $s_1 = 1$, $s_i < t_i$ for $i = 1, \ldots, \kappa - 1$, and $t_\kappa > s_\kappa$. For simplicity, we set $s_{\kappa+1} := s_1$. Let $C_i \in \mathcal{C}$ be such that $I_i$ is a $C_i$-interval. Since all faces around $v$ are covered, it holds that $s_{i+1} \leq t_i$. The face $f$ between $e_j$ and $e_{j+1}$ is covered by both, $C_i$ and $C_{i+1}$. Hence the boundary of $f$ is both a $C_i$- and a $C_{i+1}$-cycle. Thus $\langle e_{s_{i+1}}, \ldots, e_{t_i} \rangle$ is consecutive in any c-planar embedding. Hence, in any c-planar embedding, the ordering of edges around $v$ is a sequence of overlapping $C$-intervals for several $C \in \mathcal{C}$. Hence, for any face $f$ incident to $v$ there is at least one $C \in \mathcal{C}'$ such that $f$ is covered by $C$. Therefore $v$ cannot be free with respect to $\mathcal{C}'$ in any c-planar embedding.

We now apply Lemma 6 to any child cut vertex $v$ of any block $H$ and to the set $\mathcal{C}'$ of relevant clusters of any child block of $v$ to obtain our main result. Observe that the particular choice of $C_{\text{ext}}$ in the following theorem guarantees that the child cut vertices are free with respect to any relevant cluster.

We call a cut vertex $v$ free for a child block $H_i$, if $v$ is free with respect to the set of clusters that are relevant for $H_i$.

**Theorem 4** A c-connected overlapping clustered graph $(G, \mathcal{C})$ is c-planar, if and only if $G$ is planar and for each connected component of $G$, there is a root block of its BC-tree for which there exist suitable root nodes of the SPQR-tree of each block that are not inappropriate for any $C \in \mathcal{C}$ with $C_{\text{ext}} = \{v; v$ child cut vertex and $C$ relevant for a child block of $v\}$ such that

1. all binary matrices fulfill the consecutive-ones property and
2. given an arbitrary consecutive-ones ordering of the binary matrices each cut vertex is free for each of its child blocks in the corresponding embedding.

**Proof:** Assume that the two conditions hold. We embed the blocks as in the proof of Theorem 3 and combine the embeddings of the blocks as follows. Let $H$ be a block, let $v$ be a child cut vertex of $H$ and let $H_i$ be a child block of $v$. We place $H_i$ into a face of $H$ incident to $v$ that is free with respect to the set of $H_i$’s relevant clusters. This yields a c-planar embedding of $G$.

Otherwise there must be a cluster $C$ and a vertex $w \in V \setminus C$ such that $w$ is enclosed by a $C$-cycle $c$. Let $c$ be in block $H$. The first condition requires, that the embedding of $H$ is c-planar (see Theorem 3). Hence, $w$ cannot be a vertex of $H$. Let $v'$ be the parent cut vertex of $H$ and let $V'$ be the union of the sets of vertices in the connected components of $G - v'$ not containing $H$. By the choice of the root of the BC-tree, $V'$ must be drawn in the outer face of $H$. Hence $w \not\in V'$.

Finally, let $v$ be a child cut vertex of $H$, let $H_i$ be a child block of $v$, let $V_i$ be the set of vertices in the connected components of $G - v$ containing $H_i$, and assume that $w \in V_i$. Then $v$ must be enclosed by $c$ and thus, by c-planarity of $H$, $v \in C$. Since $w \not\in C$ it follows that $C$ is relevant for $H_i$. Since we embedded $H_i$ into a face of $H$ that was free with respect to $H_i$’s relevant clusters, it follows that $w$ cannot be enclosed by the $C$-cycle $c$.

For the other direction assume now that there is a c-planar embedding $E$. Without loss of generality, we assume that $G$ is connected. Let the root $H_r$ of the BC-tree be a block with an edge that is incident to the outer face of $G$. Root each SPQR-tree at an edge incident to the outer face of the respective block and incident to the parent cut vertex.

In a c-planar drawing, a child cut vertex $v$ of a block $H$ is placed on the outer face of $H[C]$ for any relevant cluster $C$ of any of $v$’s child blocks. Thus, Theorem 3 implies that the roots are not inappropriate and Condition 1 is fulfilled.

Obviously any block must be inserted into a face that is free with respect to its relevant clusters in any c-planar embedding of $G$. Consider now a block $H$ and an embedding $E'$ of $H$ corresponding to an arbitrary consecutive-ones ordering of the binary matrices. Let $v$ be a child cut vertex of $H$ and let $H_i$ be a child block of $v$. Let $C'$ be the set of relevant clusters for $H_i$.

The labeling guarantees that $v$ is on the outer face of $H[C]$ for any $C \in C'$. Thus, $E'$ is a c-planar embedding of $H$ in which $v$ is free with respect to each $C \in C'$. We further know that $E$ induces a c-planar embedding of $H$ in which $v$ is free with respect to $C'$. Hence, Lemma 6 implies that $v$ is free with respect to $C'$ in $E'$.

The characterization in the previous theorem immediately yields the following corollary.

**Corollary 1** It can be tested in polynomial time whether a c-connected overlapping clustered graph is c-planar.
Proof: We consider the connected components independently. For each choice of the root of the BC-tree and for each cluster $C$ we first proceed bottom up in the BC-tree in order to compute for each node $v$ of the BC-tree the sets $C_{\text{ext}} = \{ v; v$ child cut vertex and $C$ relevant for a child block of $v \}$ in overall $O(|V|)$ time. Then for any choice of a suitable root of the SPQR-tree of each block, we have to construct the respective matrices and check whether they have the consecutive-ones property. The size of the matrices is in $O(|V|^2|C|)$. The bottle-neck of constructing them is the handling of the $S$-nodes which can be done in $O(|V| \cdot |C|^2)$ time per $S$-node and thus in overall $O(|V|^2|C|^2)$ time. Finally, we can check whether the cut vertices are free, by simply traversing the incident faces.

Since there are $O(|V|)$ choices of the root of the BC-tree and for each SPQR-tree $O(|V|)$ choices of their roots, we can test in $O(|V|^4|C|^2)$ time whether an overlapping clustered graph is c-planar.

6 Conclusion

We showed that clustered planarity of c-connected overlapping clustered graphs can be solved in polynomial time, thus solving an open problem stated by Didimo et al. [14]. Yet, the run time of our algorithm is far from being efficient.

For special instances of the problem, there exist efficient algorithms, like the cases discussed by Didimo et al. [14] or the cases of two c-co-connected partitions discussed in Sect. 3. These approaches are based on reducing c-planarity of special subclasses of overlapping clustered graphs to c-planarity of sufficiently connected hierarchically clustered graphs. Then an efficient algorithm for this case is used, e.g., the algorithm of Cortese et al. [11].

There are two main reasons why we can not (yet) match the linear run time of this algorithm with our approach: (1) only one choice of the root of the SPQR-tree has to be considered in the hierarchical case – namely an edge that is not contained in any cluster (other than the whole vertex set), and (2) the characterization of Cortese et al. can make use of the interplay between the SPQR-tree and the rooted tree representing the cluster hierarchy. It would be interesting whether these ideas could be extended to our case, as an interplay between the SPQR-tree and the Hasse diagram.
References


