Advances in C-Planarity Testing of Clustered Graphs (Extended Abstract)

Carsten Gutwenger[2], Michael Jünger[1], Sebastian Leipert[2], Petra Mutzel[3], Merijam Percan[1], and René Weiskircher[3]

[1] Universität zu Köln, Institut für Informatik,
Pohligstraße 1, 50969 Köln, Germany
{mjuenger,percan}@informatik.uni-koeln.de
Partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT).
[2] caesar research center,
Friedensplatz 16, 53111 Bonn, Germany
{gutwenger,leipert}@caesar.de
[3] Technische Universität Wien E186, Favoritenstraße 9–11, 1040 Wien, Austria,
{mutzel,weiskircher}@ads.tuwien.ac.at

Abstract. A clustered graph $C = (G, T)$ consists of an undirected graph G and a rooted tree T in which the leaves of T correspond to the vertices of $G = (V, E)$. Each vertex c in T corresponds to a subset of the vertices of the graph called "cluster". c-planarity is a natural extension of graph planarity for clustered graphs, and plays an important role in automatic graph drawing. The complexity status of c-planarity testing is unknown. It has been shown in [FCE95,Dah98] that c-planarity can be tested in linear time for c-connected graphs, i.e., graphs in which the cluster induced subgraphs are connected.
In this paper, we provide a polynomial time algorithm for c-planarity testing for "almost" c-connected clustered graphs, i.e., graphs for which all c-vertices corresponding to the non-c-connected clusters lie on the same path in T starting at the root of T, or graphs in which for each non-connected cluster its super-cluster and all its siblings are connected. The algorithm uses ideas of the algorithm for subgraph induced planar connectivity augmentation presented in [GJL$^+$02]. We regard it as a first step towards general c-planarity testing.

1 Introduction

A clustered graph consists of a graph G and a recursive partitioning of the vertices of G. Each partition is a cluster of a subset of the vertices of G. Clustered graphs are getting increasing attention in graph drawing [BDM02,EFN00,FCE95,Dah98]. Formally, a *clustered graph* $C = (G, T)$ is defined as an undirected graph G and a rooted tree T in which the leaves of T correspond to the vertices of $G = (V, E)$.

In a cluster drawing of a clustered graph, vertices and edges are drawn as usual, and clusters are drawn as simple closed curves defining closed regions of the plane. The region of each cluster C contains the vertices W corresponding to C and the edges of the graph induced by W. The borders of the regions for the clusters are pairwise disjoint. If a cluster drawing does not contain crossings between edge pairs or edge/region pairs, we call it a *c-planar* drawing. Graphs that admit such a drawing are called *c-planar*.

While the complexity status of c-planarity testing is unknown, the problem can be solved in linear time if the graph is c-connected, i.e., all cluster induced subgraphs are connected [Dah98,FCE95]. In approaching the general case, it appears natural to augment the clustered graph by additional edges in order to achieve c-connectivity without loosing c-planarity.

The results presented in this paper are the basis for a first step towards this goal. Namely, we present a polynomial time algorithm that tests c-planarity for "almost" c-connected clustered graphs, i.e., graphs for which all c-vertices corresponding to the non-connected clusters lie on the same path in T starting at the root of T, or graphs in which for each non-connected cluster its super-cluster and all its siblings are connected.

The algorithm uses ideas from the linear time algorithm for subgraph induced planar connectivity augmentation presented in [GJL$^+$02]. For an undirected graph $G = (V, E)$, $W \subseteq V$, and $E_W = \{(v_1, v_2) \in E : \{v_1, v_2\} \subseteq W\}$ let $G_W = (W, E_W)$ be the subgraph of G induced by W. If G is planar, a *subgraph induced planar connectivity augmentation* for W is a set F of additional edges with end vertices in W such that the graph $G' = (V, E \cup F)$ is planar and the graph G'_W is connected.

The paper is organized as follows. After an introduction into the SPQR data structure and clustered graphs in Section 2, we describe in Section 3 a linear time algorithm for c-planarity testing of a clustered graph with exactly one cluster in addition to the root cluster. This algorithm can be extended to a quadratic time algorithm for c-planarity testing in clustered graphs with one level beyond the root level in which at most one cluster may be non-connected (see Section 4). In Section 5 we suggest a technique to extend the previous results to graphs with arbitrarily many non-connected clusters. The only requirement is that for each non-connected cluster, all its siblings and its super-cluster are connected. The same technique can be applied for graphs in which all the non-connected clusters lie on the same path in T.

2 Preliminaries

SPQR-trees have been suggested by Di Battista and Tamassia [BT96]. They represent a decomposition of a planar biconnected graph according to its split pairs (pairs of vertices whose removal splits the graph or vertices connected by an edge). The construction of the SPQR-tree works recursively. At every node v of the tree, we split the graph into the split components of the split pair associated with that node. The first split pair of the decomposition is an edge of the graph and is called the *reference edge* of the SPQR-tree. We add an edge to each of them to make sure that they are biconnected and continue by computing their SPQR-tree and making the resulting trees the subtrees of the node used for the splitting. Every node of the SPQR-tree has two associated graphs:

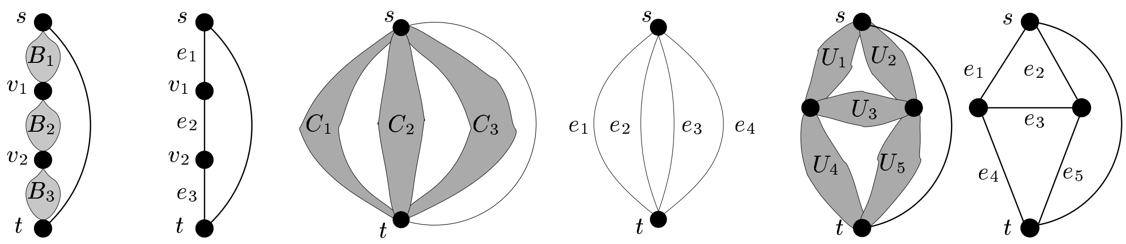
- The *skeleton* of the node associated with a split pair p is a simplified version of the whole graph where some split-components are replaced by single edges.
- The *pertinent graph* of a node v is the subgraph of the original graph that is represented by the subtree rooted at v.

The two vertices of the split pair that are associated with a node v are called the *poles* of v. There are four different node types in an SPQR-tree (S-,P-,Q- and R-nodes) that differ in the number and structure of the split components of the split pair associated with the node. The Q-nodes form the leaves of the tree, and their is one Q-node for each edge in the graph. The skeleton of a Q-node consists of the poles connected by two edges. The skeletons of S-nodes are cycles, while the skeletons of R-nodes are triconnected graphs. P-node skeletons consist of the poles connected by at least three edges. Figure 1 shows examples for skeletons of S-, P- and R-nodes.

Skeletons of adjacent nodes in the SPQR-tree share a pair of vertices. In each of the two skeletons, one edge connecting the two vertices is associated with a corresponding edge in the other skeleton. These two edges are called *twin edges*. The edge in a skeleton that has a twin edge in the parent node is called the *virtual edge* of the skeleton.

Definition 1 (Expansion Graph). *Let $e = (u, v)$ be an edge in a skeleton $\mathcal{S}$ of a node μ of the SPQR-tree $\mathcal{T}$ of G. Since e is an edge in a skeleton, the vertices u and v are a split-pair of G. Then we define the* expansion graph *of e as follows:*

1. *If μ is the Q-node for edge e' in G, then the expansion graph $G(e)$ of e is defined as follows: if e is the virtual edge of μ then we define $G(e)$ as $G(e) = (V, (E - \{e'\}) \cup \{e\})$. Otherwise, we define $G(e)$ as $G(e) = (\{u, v\}, \{e, e'\})$. Therefore the expansion graph is either isomorphic to G or to $\mathcal{S}$.*
2. *If μ is an R-node or S-node, then the expansion graph of e is the union of all the split components of the split pair $\{u, v\}$ in G that contain no vertices of $\mathcal{S}$ except u and v together with edge e.*



(a) Case S-node (b) Case P-node (c) Case R-node

Fig. 1. The structure of biconnected graphs and the skeleton of the root of the corresponding SPQR-tree.

3. *If μ is a P-node, then there are at least three split components of the pair $\{u, v\}$ in G. In the construction of $\mathcal{T}$, all edges e_i of $\mathcal{S}$ except the virtual edge are associated with a subgraph G_i of G. Thus we define the expansion graph of each e_i as the graph G_i together with edge e. The expansion graph of the virtual edge is defined as the split component of $\{u, v\}$ that contains the reference edge of $\mathcal{T}$ (the edge of G that is used to start the decomposition) together with edge e.*

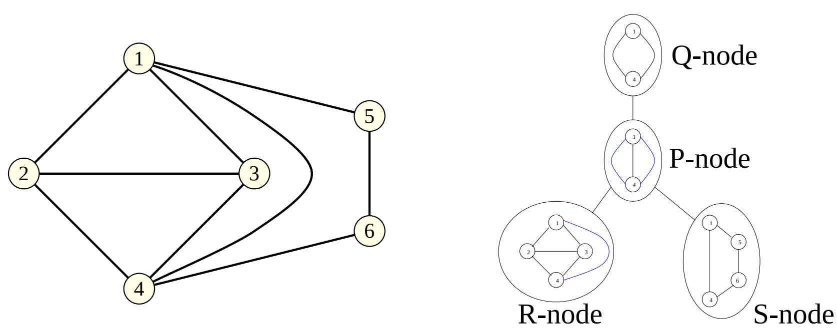


Fig. 2. A graph G and its SPQR-tree (the Q-nodes of the R- and S-node are omitted).

All leaves of the SPQR-tree are Q-nodes and all inner nodes S-, P- or R-nodes. When we see the SPQR-tree as an unrooted tree, then it is unique for every biconnected planar graph. Another important property of these trees is that their size (including the skeletons) is linear in the size of the original graph and that they can be constructed in linear time [BT96,GM01]. As described in [BT96] [GM01], SPQR-trees can be used to represent the set of all combinatorial embeddings of a biconnected planar graph. Every combinatorial embedding of the original graph defines a unique combinatorial embedding for each skeleton of a node in the SPQR-tree. Conversely, when we define an embedding for each skeleton of a node in the SPQR-tree, we define a unique embedding for the original graph. The skeleton of S- and Q-nodes are simple cycles, so they have only one embedding. But the skeletons of R-and Q-nodes have at least two different embeddings. Therefore, the embeddings of the R- and P-nodes determine the embedding of the graph and we call these nodes the *decision nodes* of the SPQR-tree. The *BC-tree* of a connected graph has two types of nodes: The c-nodes correspond to cut-vertices of G and the b-nodes to biconnected components (blocks). There is an edge connecting a c-node and a b-node, if the cut-vertex is contained in the block.

2.1 Clustered Graphs

The following definitions are based on the work of Cohen, Eades and Feng [FCE95]. A *clustered graph* $C = (G,T)$ consists of an undirected graph G and a rooted tree T where the leaves of T are the vertices of G. Each node ν of T represents a *cluster* $V(\nu)$ of the vertices of G that are leaves of the subtree rooted at ν. Therefore, the tree T describes an inclusion relation between clusters. T is called the *inclusion tree* of C, and G is the *underlying graph* of C. The root of T is called *root cluster*. We let $T(\nu)$ denote the subtree of T rooted at node ν and $G(\nu)$ denote the subgraph of G induced by the cluster associated with node ν. We define $C(\nu) = (G(\nu), T(\nu))$ to be the *sub-clustered graph* associated with node (ν). A *drawing* of a clustered graph $C = (G,T)$ is a representation of the clustered graph in the plane. Each vertex of G is represented by a point. Each edge of G is represented by a simple curve between the drawings of its endpoints. For each node ν of T, the cluster $V(\nu)$ is drawn as a simple closed region R that contains the drawing of $G(\nu)$, such that:

- the regions for all sub-clusters of R are completely in the interior of R;
- the regions for all other clusters are completely contained in the exterior of R;
- if there is an edge e between two vertices of $V(\nu)$ then the drawing of e is completely contained in R.

We say that there is an *edge-region crossing* in the drawing if the drawing of edge e crosses the drawing of region R more than once. A drawing of a clustered graph is *c-planar* if there are no edge crossings or edge-region crossings. If a clustered graph C has a c-planar drawing then we say that it is *c-planar* (see Figure 3). Therefore, a c-planar drawing contains a planar drawing of the

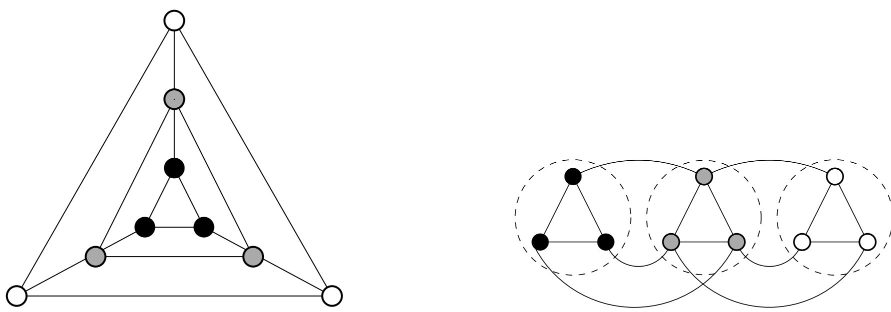


Fig. 3. A planar clustered graph that is not c-planar [FCE95] (the three disjoint clusters are represented by different types of vertices).

underlying graph. An edge is said to be *incident* to a cluster $V(\nu)$ if one end of the edge is a vertex of the cluster but the other endpoint is not in $V(\nu)$. An *embedding* of C includes an embedding of G plus the circular ordering of edges crossing the boundary of the region of each non trivial cluster (a cluster which is not a single vertex). A clustered graph $C = (G,T)$ is connected if G is connected. A clustered graph $C = (G,T)$ is a *c-connected* clustered graph if each cluster induces a connected subgraph of G. Suppose that $C_1 = (G_1, T_1)$ and $C_2 = (G_2, T_2)$ are two clustered graphs such that T_1 is a subtree of T_2, and for each node ν of T_1, $G_1(\nu)$ is a subgraph of $G_2(\nu)$. Then we say C_1 is a *sub-clustered graph* of C_2, and C_2 is a *super-clustered graph* of C_1. The following results from [FCE95] characterize c-planarity:

Theorem 1. *[FCE95] A c-connected clustered graph $C = (G,T)$ is* c-planar *if and only if graph G is planar and there exists a planar drawing $\mathcal{D}$ of G, such that for each node ν of T, all the vertices and edges of $G - G(\nu)$ are in the outer face of the drawing of $G(\nu)$.*

Theorem 2. *[FCE95] A clustered graph $C = (G,T)$ is c-planar if and only if it is a sub-clustered graph of a connected and c-planar clustered graph.*

A further result from [FCE95] is a c-planarity testing algorithm for c-connected clustered graphs based on Theorem 1 with running time $O(n^2)$, where n is the number of vertices of the underlying graph and each non-trivial cluster has at least two children. An improvement in time complexity is given by Dahlhaus who constructed a linear time algorithm [Dah98].

3 Clustered Graphs with Two Clusters

Let $C = (G, T)$ be a clustered graph with a root cluster and a cluster ν. Let the graph G be connected and the subgraph induced by the vertices of the cluster not connected. The problem of connecting the subgraph induced by one cluster is similar to the problem of planar connectivity augmentation of an induced subgraph [GJL+02].

In the following, we call the vertices belonging to ν *blue vertices*. We assign one of two colors to each edge in each skeleton: blue or black. We call an edge in a skeleton blue, if its expansion graph contains blue vertices and black otherwise.

Additionally, we assign the attribute *permeable* to some blue edges. Intuitively, an edge is permeable if we can construct a path connecting only blue vertices through its expansion graph. Let $G(e)$ be the expansion graph of edge e in skeleton $\mathcal{S}$. In any planar embedding $G(e)$, there are exactly two faces that have e on their boundary. This follows from the fact that in a planar biconnected graph, every edge is on the boundary of exactly two faces in every embedding. We call the edge e in $\mathcal{S}$ permeable with respect to W, if there is an embedding Π of $G(e)$ and a list of at least two faces $L = (f_1, \ldots, f_k)$ in Π that satisfies the following properties:

1. The two faces f_1 and f_k are the two faces with e on their boundary.
2. For any two faces f_i, f_{i+1} with $1 \leq i < k$, there is a blue vertex on the boundary between f_i and f_{i+1}.

We call a skeleton $\mathcal{S}$ of a node v of $\mathcal{T}$ permeable if the pertinent graph of v and the virtual edge of $\mathcal{S}$ have the two properties stated above. So $\mathcal{S}$ is permeable if the twin edge of its virtual edge is permeable.

Theorem 3. *Let $C = (G, T)$ be a clustered graph where G is a connected planar graph and W is the vertex set corresponding to the only non-trivial non-root cluster in T. We further assume that the subgraph G_W induced by W is connected. The clustered graph C is c-planar if and only if there is an embedding of G that contains no cycle of blue vertices which separates the black vertices.*

Theorem 4. *A connected clustered graph $C = (G, T)$ where G is planar with one non-connected cluster ν is c-planar if and only if*

1. *all P-nodes of its SPQR-tree contain at the most two blue but not permeable edges,*
2. *there exists a planar connectivity augmentation of the subgraph induced by the non-connected cluster ν and*
3. *there is an embedding of G that contains no circle of blue vertices that separates black vertices.*

Hence we have to change the algorithm for the subgraph induced planar connectivity augmentation in the following way:

First, we transform the not c-connected clustered graph into a c-connected clustered graph.

Theorem 5. *Let C be a clustered graph with a connected root cluster and a non-connected cluster ν. Let C_{sub} be its sub-clustered graph created by splitting ν into one dummy cluster for each connected component of the subgraph induced by ν that contain at least two vertices (see Figure 4). C is not c-planar if C_{sub} is not c-planar. C_{sub} is a c-connected clustered graph. We call C_{sub} the c-split clustered graph of C.*

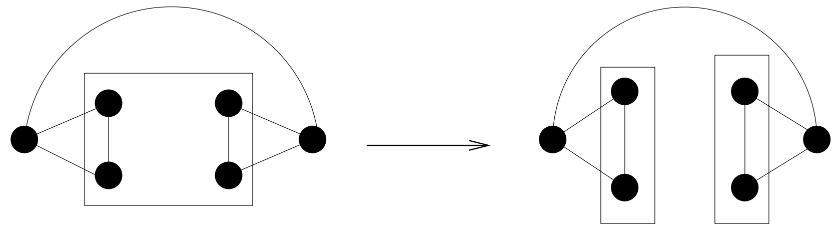


Fig. 4. A clustered graph C with a non-connected cluster. By splitting the non-connected cluster, C is extended to a c-connected clustered graph.

Proof. We assume that C_{sub} is not c-planar. Therefore there exists an edge-crossing or a cluster-crossing in at least one cluster ν. The subgraphs induced by the dummy clusters of C_{sub} that are not the root cluster are connected components of the corresponding non-connected cluster μ of C. Hence there has to exist an edge-crossing or a cluster-crossing in μ. As a result, C cannot be c-planar.

We add a dummy cluster to each connected component of the non-connected cluster ν that has at least two vertices making it the child cluster of ν. Then we delete ν and we obtain a c-connected clustered graph C'. Then we test c-planarity of C' using the c-planarity testing by Dahlhaus [Dah98]. In the positive case, we apply the planar connectivity augmentation algorithm on G and the vertices of ν as subset W.

Next we consider the case that G is non-connected and there is only one cluster ν that is not the root cluster and is non-connected. For all connected components of G we apply our algorithm for clustered graphs with one non-connected cluster. Then we choose for each connected component a face as outer face that contains at least one blue vertex v_1 and one non-blue vertex v_2 so that the **l**owest **c**ommon **a**ncestor (lca) of v_1 and v_2 is the root cluster. Then we connect them in the outer face so that the edges of a minimum cardinality edge set are inserted (as described in the subgraph induced planar connectivity augmentation algorithm for non-connected graphs [GJL+02]). It is obvious that c-planarity of a clustered graph C that has a root cluster and one non-connected non-trivial cluster ν can be tested in linear time. In the positive case a c-planar embedding with a minimum cardinality augmenting edge set will be computed. Then we say that the clustered graph has a c-planar connectivity augmentation for the non-connected cluster ν.

Algorithm 1: The algorithm for clustered graphs $C = (G, T)$ that contain a connected root cluster and a non-connected cluster ν. It computes an embedding Π and the minimum cardinality augmenting edge set.

Input: A clustered graph $C = (G, T)$ that contains a connected root cluster and a non-connected cluster ν.

Result : `true` if and only if there is a c-planar connectivity augmentation for ν; in the positive case an embedding Π and the minimum cardinality augmenting edge set will be computed.

Copy C and modify the copy into a c-connected clustered graph C' by splitting the non-connected cluster for each connected component of the subgraph $G(\nu)$;

Apply the linear time c-planarity test on C';

if the test return false **then**

    `return false`;

Apply the subgraph induced planarity augmentation algorithm for C and ν;

4 One-Level Clustered Graphs

As shown in the last section we can test c-planarity of a clustered graph with one non-connected clustered graph $C = (G, T)$ and a root cluster in $O(n)$ time, where n is the number of vertices of G. Next, we consider a clustered graph with a cluster tree T with only one level except the root cluster. Further let only one child cluster of the root cluster be non-connected. We assume, that each non-trivial cluster (that is a cluster that is not a leaf) has at least two children. Until now, we

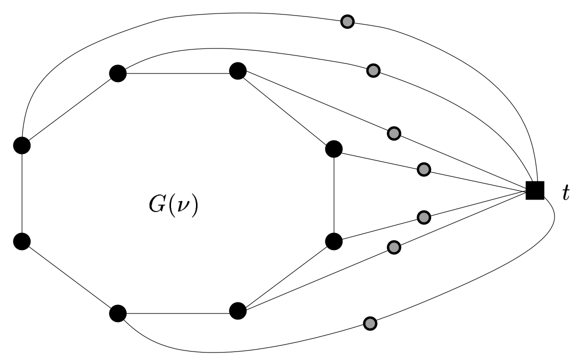


Fig. 5. Constructing an auxiliary graph $G'(\nu)$ from the connected subgraph $G(\nu)$ where the incident edges of ν are connected with a dummy vertex t [FCE95].

cannot test c-planarity of a general clustered graph, a clustered graph with more then two clusters has to be c-connected to test c-planarity.

We can construct an auxiliary c-connected graph C of a general clustered graph by splitting the clusters that are non-connected as mentioned in the last section. We assume that the graph G of the clustered graph $C = (G, T)$ is connected, otherwise we can apply this technique to each connected component of G. We modify a copy of C to get the c-split clustered graph of C and to obtain a c-connected auxiliary clustered graph C'. We test c-planarity of C' with the c-planarity test by Cohen, Eades and Feng and in the positive case we extend the representative graph G_{PQ} of the graph G in the last step of the c-planarity test algorithm with the planar connectivity algorithm if G_{PQ} is planar.

The representative graph G_{PQ} contains a wheel graph for each biconnected component of G. Cut vertices of G correspond to cut vertices of G_{PQ} and every vertex in G_{PQ} has its counterpart in G except the vertices constructed as the hubs of wheels (see Figure 6). Hence, the representative graph represents all possible orderings of edges that are incident to a cluster ν around the outer face of $G(\nu)$. Therefore the clustering structure is represented in G_{PQ}.

Finally, we apply the planar connectivity augmentation algorithm for the vertices of the non-connected cluster ν of G_{PQ}. The vertices on the rim face of a wheel graph correspond to vertices of the graph G. We can create the c-split clustered graph C_{sub} in $O(n)$ time where n is the number of vertices of G. The c-planarity testing can be done in $O(n^2)$ time and the planar connectivity augmentation of the subgraph induced by the non-connected clustered graph in $O(n)$ time. As a result the algorithm can be implemented in $O(n^2)$ time where n is the number of vertices of G.

5 Multi-Level Clustered Graphs

We extend our ideas from the previous section to clustered graphs with more than one level. Next we consider a clustered graph $C = (G, T)$ with at least two non-connected clusters. We assume that the root cluster is connected that means that G is connected. If for every non-connected

Algorithm 2: The algorithm for clustered graphs $C = (G, T)$ that contain a connected root cluster, a non-connected child cluster ν and an arbitrarily number of connected child clusters. It computes an embedding Π and the minimum cardinality augmenting edge set if C is c-planar.

Input: A clustered graph $C = (G, T)$ that contains a connected root cluster, a non-connected child cluster ν and an arbitrarily number of connected child clusters.

Result : `true` if and only if there is a c-planar connectivity augmentation for ν; in the positive case an embedding Π and the minimum cardinality augmenting edge set will be computed.

Copy C and modify the copy to get a c-connected clustered graph C' by splitting the non-connected cluster for each connected component of the subgraph $G(\nu)$;
Apply the c-planarity test by Cohen, Eades and Feng on C';
if the test return false **then**
  `return false`;
Apply on the representative graph G_{PQ} of the last step in the c-planarity testing the subgraph induced planarity augmentation algorithm for C and ν;

cluster ν in the cluster tree T there exists a connected parent cluster and if all siblings of ν are connected, we will be able to connect the non-connected clusters using the planar connectivity augmentation [GJL$^+$02].

First, we modify a copy of C to get a c-split clustered graph of C. Then, we proceed from the leaves to the root of the cluster tree T. For every non-connected cluster ν that has connected siblings μ and a connected parent $pa(\nu)$, we have to test whether the induced subgraphs are planar, and whether the edges that are incident to the clusters can be drawn into the outer face of the induced subgraphs. Therefore we have to take the incident edges of each cluster into account.

We extend the c-planarity test by Cohen, Eades and Feng as follows: For every connected parent cluster $pa(\nu)$ with a non-connected child cluster ν, we form a graph $G'(pa(\nu))$ by adding the incident edges of $pa(\nu)$ to $G(pa(\nu))$ as constructed in the c-planarity test (see Figure 5). We add a dummy vertex on each incident edge of $pa(\nu)$ and connect the dummy vertices with another dummy vertex t. Then we apply the linear time planarity test based on PQ-trees [BL76,CEL67,HT74] to $G'(pa(\nu))$ and in the positive case the planar connectivity augmentation for the subgraph induced by the non-connected cluster [GJL$^+$02]. As we do this recursively for each connected parent node

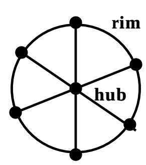


Fig. 6. A wheel graph.

which has a non-connected cluster we can test c-planarity in $O(n^2)$ time where n is the number of vertices of G.

Furthermore, we can apply our technique to connected clustered graphs $C = (G, T)$ where the non-connected clusters lie on the same path from the root. We modify a copy of C to obtain a c-split clustered graph $C' = (G, T')$. We start from the leaves of the cluster tree T' and proceed level by level to the root cluster. We apply the same technique as before building the representative graphs. The only difference is that we apply the planar connectivity augmentation algorithm for every non-connected cluster using the representative graph G_{PQ} that is constructed when all nodes of T' are traversed.

Then we start again, but now we traverse the path of non-connected clusters in the cluster tree T from the leaves to the root and apply for the vertices of these clusters the planar connectivity

Algorithm 3: The algorithm for clustered graphs that contain a connected root cluster and non-connected clusters where its parent cluster and its sibling clusters are connected. It computes an embedding Π and the minimum cardinality augmenting edge set if C is c-planar.

Input: A clustered graph $C = (G, T)$ that contains a connected root cluster and a non-connected cluster ν if and only if the its parent cluster and its sibling cluster are connected.

Result : `true` if and only if there is a c-planar connectivity augmentation for ν; in the positive case an embedding Π and the minimum cardinality augmenting edge set will be computed.

Copy C and modify the copy into a c-connected clustered graph C' by splitting the non-connected cluster for each connected component of the subgraph $G(\nu)$;
Change the c-planarity test of Cohen, Eades and Feng as follows and apply it to C';
for the leaves to the root cluster of the cluster tree T **do**
  Construct for the cluster ν the graph $G'(\nu)$ from the subgraph $G(\nu)$ as in the c-planarity test of Cohen, Eades and Feng;
  Test planarity on $G'(pa(\nu))$ as in the c-planarity test of Cohen, Eades and Feng;
  if the planarity test returns false **then**
    return false;
  Apply the subgraph induced planarity augmentation algorithm for the vertices of all dummy clusters ν in $G'(pa(\nu))$ if $pa(\nu)$ is an original cluster of T;

augmentation algorithm [GJL+02] in G. This can be done in $O(n^2)$ time where n is the number of vertices in the underlying graph G.

Finally, we consider two non-connected clusters that are siblings in an arbitrary clustered graph where all other clusters are connected and G is planar and connected. If the two clusters are contained in two different connected components or if they are contained in two different biconnected components or in two different subtrees of a BC-tree, we can apply the one-cluster-method for each connected or biconnected component independently. This can be extended to an arbitrarily number of non-connected clusters that are siblings under the condition that they are in different connected or biconnected components.

Hence, there is a large class of clustered graphs that can now be tested for c-planarity.

References

[BDM02] G. Di Battista, W. Didimo, and A. Marcandalli. Planarization of clustered graphs (extended abstract). In P. Mutzel, Michael Jünger, and Sebastian Leipert, editors, *Graph Drawing*, volume 2265 of *Lecture Notes in Computer Science*, pages 60–74. Springer-Verlag, 2002.

[BL76] K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences*, 13(1):335–379, 1976.

[BT96] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM Journal on Computing*, 25(5):956–997, 1996.

[CEL67] I. Cederbaum, S. Even, and A. Lempel. An algorithm for planarity testing of graphs. *In Theory of Graphs, International Symposium, Rome*, pages 215–232, 1967.

[Dah98] E. Dahlhaus. Linear time algorithm to recognize clustered planar graphs and its parallelization (extended abstract). In C. L. Lucchesi, editor, *LATIN '98, 3rd Latin American symposium on theoretical informatics, Campinas, Brazil, April 20–24, 1998.*, volume 1380 of *Lecture Notes in Computer Science*, pages 239–248, 1998.

[EFN00] P. Eades, Q.-W. Feng, and H. Nagamochi. Drawing clustered graphs on an orthogonal grid. *Journal of Graph Algorithms and Applications*, 3:3–29, 2000.

[FCE95] Q.-W. Feng, R.-F. Cohen, and P. Eades. Planarity for clustered graphs. In P. Spirakis, editor, *Algorithms – ESA '95, Third Annual European Symposium*, volume 979 of *Lecture Notes in Computer Science*, pages 213–226. Springer-Verlag, 1995.

[GJL+02] C. Gutwenger, M. Jünger, S. Leipert, P. Mutzel, M. Percan, and R. Weiskircher. Subgraph induced planar connectivity augmentation. Technical report, Institut für Informatik, Universität zu Köln, 2002. in preparation.

[GM01] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In J. Marks, editor, *Graph Drawing (Proc. 2000)*, volume 1984 of *Lecture Notes in Computer Science*, pages 77–90. Springer-Verlag, 2001.

[HT74] J. Hofcroft and R. E. Tarjan. Efficient planarity testing. *Journal of ACM*, 21(4):549–568, 1974.