Triangulating Clustered Graphs

Michael Jünger[1], Sebastian Leipert[2], and Merijam Percan[1]

[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
{leipert}@caesar.de

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 μ 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. As we triangulate a planar embedded graph so that G is still planar embedded after triangulation, we consider triangulation of a c-connected clustered graph that preserve the c-planar embedding.
In this paper, we provide a linear time algorithm for triangulating c-connected c-planar embedded clustered graphs $C = (G, T)$ so that C is still c-planar embedded after triangulation. We assume that every non-trivial cluster in C has at least two childcluster. This is the first time, this problem was investigated.

1 Introduction

A planar graph is triangulated (triangular or maximal planar) when every face has exactly three vertices. If a planar graph is not triangulated, then there exists a face f that has at least four different vertices, for example v_1, v_2, v_3 and v_4 in this order around the face. Hence G is planar, we can achieve a planar embedding. Because of that, the edges (v_1, v_3) and (v_2, v_4) are not included in f. Adding one of the edge and repeating this for all not triangulated faces, we receive a triangulated planar graph. It is assumed, that G is biconnected planar, otherwise an augmentation algorithm is used. The first triangulation algorithm is due to Read [Rea87], and modified by De Freysseix so that it works in linear space. It works as follows: All vertices are visited. For every pair of consecutive neighbors u and v of a current visited vertex v, we add an edge $e = (u, w)$ to G according the planar embedding if w is not adjacent to u. Applying this to all vertices of G a triangulated graph is received that can contain multiple edges. The multiple edges can be replaced as follows: deleting the multiple edge we get a face with four vertices. Introducing an edge for the other two vertices that are not end vertices of the deleted multiple edge, one multiple edge is deleted. Hagerup and Uhrig [HU91] modified that algorithm so that no multiple edges are introduced. Kant presents in [Kan93] a canonical triangulation as a good and simple method for computing a canonical ordering while triangulating the graph. He also give a proof of NP-completeness of triangulation of planar graphs while minimizing the maximum degree and introduces an approximation algorithm. Therefore, the problem of triangulating planar graphs is well studied. 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,GJL$^+$02b,GJL$^+$02a]. 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 $G = (V, E)$. G is called the underlying graph of C.

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 first algorithms that uses triangulation of clustered graphs is presented in [EFN00] where the underlying graph G is triangulated. We show in the paper, that triangulating clustered graphs by triangulating the underlying graph G without taking the clustering structure can cause problems. As we know, this is the first time that this problem is investigated. We present in this paper a linear time algorithm to prepare a c-connected clustered graph for triangulating the underlying graph G without loosing the c-planar embedding.

2 Preliminaries: Clustered Graph

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*. A cluster of T is a *trivial cluster* if and only if it is a leaf of T. Each leaf of T is a vertex of G.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. We define $pa(\nu)$ the parent cluster of ν in T and $chl(\nu)$ the set of child clusters of ν in T. 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 1). Therefore, a c-planar drawing contains a planar drawing of the

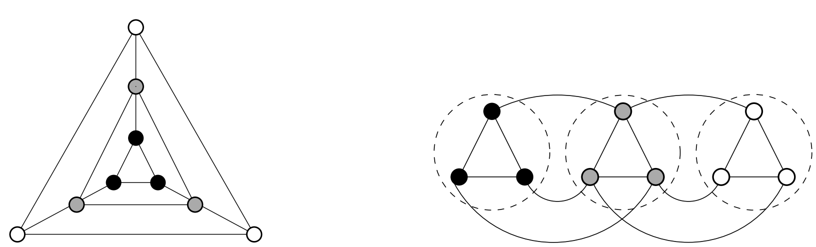

Fig. 1. A planar clustered graph that is not c-planar [FCE95] (the three disjoint clusters are represented by different types of vertices)

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 *c-connected* 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].

We assume in this paper that all non-trivial clusters of clustered graphs have at least two child clusters.

3 Triangulation of clustered graphs as same as of graphs?

The triangulations of graphs mentioned in the introduction preserve the planar embedding of a graph.

Consider a clustered graph with a c-planar embedding. We expect as a condition of a triangulation of a clustered graph C that C is still c-planar after triangulation.

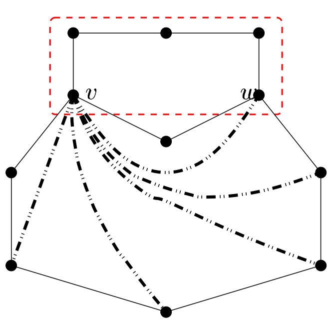

Fig. 2. C-Planar embedded clustered graph: triangulation destroys c-planar embedding

In Figure 2 there is a c-connected clustered graph $C = (G, T)$ with a c-planar embedding triangulated in one face by the common way. This means that we triangulate its underlying graph G. We notice, that the edge $e = (v, w)$ with end vertices in cluster ν isolates a vertex that belongs to the root cluster. Therefore, in this example, C looses its c-planar embedding after triangulation. Hence, by triangulation of the whole graph G, C can loose its c-planarity. This is so because the common triangulation do not take the cluster structure into account. Note that applications on triangulated c-connected clustered graphs with a c-planar embedding as the c-st-numbering constructed in [EFN00] must take this into account. If we triangulate the c-connected c-planar embedded clustered graph with the common way, the c-st-numbering can fail. This is so because the auxiliary graph computed for each level of the cluster tree T that is received by shrinking each cluster in one node in the algorithm of the c-st-numbering does not guarantee biconnectivity if G was triangulated as common.

As we consider general c-planar embedded clustered graphs, Figure 3 shows that triangulating only the underlying graph G can destroy the c-planar embedding and the c-planarity.

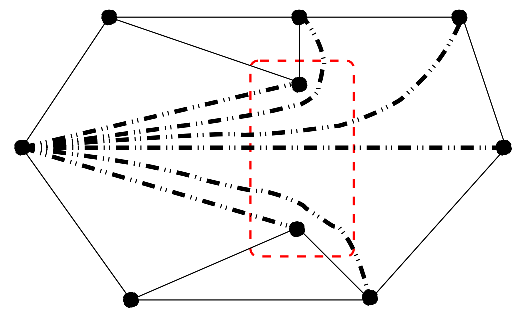

Fig. 3.

That is why we cannot extend clustered graphs in the sense of extending only graph G. It is significant to consider the clustering structure. In the next sections we show solutions of this problem.

4 C-Triangulation of C-Connected Clustered Graphs

First we consider c-connected clustered graphs $C = (G, T)$ with a c-planar embedding. In the previous section we mentioned that in Figure 2 the edge $e = (v, w)$ with end vertices in the cluster ν not equal to the root cluster isolates a vertex that belongs only to the root cluster. Hence the subgraph $G - G(\nu)$ is non-connected and has at least two connected components. If we isolate the connected components by edges introduced by triangulating G, we loose the c-planar embedding of C. Hence we have to construct a triangulation that prevents isolations. Before we discuss the solution, we note that the following follows immediately: If in Figure 2 the subgraph $G - G(\nu)$ would be connected, then no isolation would occur by triangulating the underlying graph G. This leads us to the following theorem.

Theorem 3. *Let $C = (G, T)$ be a c-connected clustered graph with a c-planar embedding and for all non-trivial clusters ν of T, $G - G(\nu)$ is connected. If C is triangulated by triangulating the underlying graph G without triangulating the outer face, C is still c-planar embedded after triangulation.*

Proof. A c-connected clustered graph is c-planar if G is planar and for all clusters ν, $G - G(\nu)$ can be drawn outside of the drawing of $G(\nu)$ (see Theorem 1). If additionally $G - G(\nu)$ is connected for all clusters ν, it is obvious that $G - G(\nu)$ can be drawn outside of $G(\nu)$ for all clusters ν. As we add edges to C, we do not destroy the properties of C. Therefore, for all clusters ν, $G(\nu)$ is still connected, and $G - G(\nu)$ is also still connected. Therefore, C is still c-planar. As mentioned in [GJL$^+$02b,GJL$^+$02a], we destroy the c-planar embedding if we introduce edges to a connected cluster μ of T that isolates vertices belonging to a cluster ν not equal to μ. But this only happens, if one of the following cases is true:

1. Either ν is an ancestor of μ, but then $G - G(\mu)$ is not connected before augmentation, or
2. ν is no ancestor of μ, but then $G(\nu)$ is not connected before augmentation.

If $G(\nu)$ is connected, we destroy the c-planar embedding, if we isolate $G(\nu)$ by adding edges to a connected cluster μ not descendant of ν around the drawing of $G(\nu)$. But this only happens if a connected component of $G - \{G(\nu) \cup G(\mu)\}$ is isolated to $G(\nu)$. But this means that $G - G(\mu)$ is not connected in the c-planar embedding before adding the edges. □

Now we have to consider a c-connected clustered graph $C = (G, T)$ with a c-planar embedding in which at least one non-trivial cluster ν of T exists for which $G - G(\nu)$ is non-connected.

We define $chl(\nu)$ as the set of child clusters of ν and $chl_{nt}(\nu)$ as the set of nontrivial child clusters.

Definition 1. *Let F be the subset of faces of a c-planar embedded clustered graph $C = (G, T)$. If there is a node (an edge) that belongs to the surroundings of a face f of F, we write $v \in f$ $(e \in f)$.*

$G - G(\nu)$ is non-connected means that by deleting $G(\nu)$ the clustered graph splits in at least two connected components.

Definition 2. *Let $C = (G, T)$ be a c-connected clustered graph. If there is no cluster ν so that C splits into at least two connected components by deleting $G(\nu)$ we say that C is cluster-biconnected. If such a cluster exists, we call ν a cluster-cut-vertex and C cluster-connected.*

Therefore, by Theorem 3: C-planar embedded *cluster*-biconnected clustered graphs C can be triangulated by triangulating G without loosing the c-planar embedding of C. As we look at the proof of Theorem 3 we see that it is obvious that *cluster*-biconnected clustered graphs are c-planar if and only if G is planar. Therefore, *cluster*-biconnected clustered graphs can be triangulated by triangulating the planar graph G. Therefore, *cluster*-biconnected c-planar clustered graphs can be augmented by augmenting G. Therefore, all augmentations known for graphs can be applied on planar *cluster*-biconnected clustered graphs, they depend only on the properties of G. We have to take the outer face into account by modifying C. We will discuss this later in this paper. Note that the c-connected clustered graph C can be *cluster*-biconnected if the underlying graph G is not biconnected (see Figure 4). The clustered graph in Figure 2 is not *cluster*-biconnected.

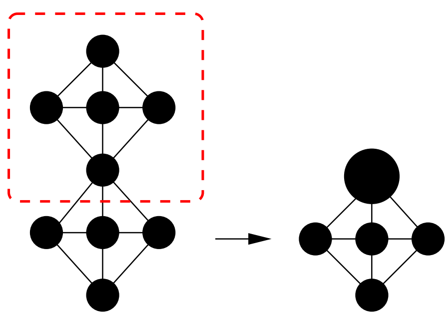

Fig. 4. Left: A *cluster*-biconnected clustered graph $C = (G, T)$ where G is not biconnected. Right: The graph G_{shrink} of the root cluster.

Now we have to consider, how we can augment c-planar embedded c-connected clustered graphs to make them *cluster*-biconnected. The idea is to modify the c-planar embedded c-connected clustered graph into graphs $G_{shrink}(\nu)$ for each cluster ν in the way that cut vertices in $G_{shrink}(\nu)$ correspond to *cluster*-cut-vertices in $C(\nu)$. We do this by shrinking the clusters of C according to the c-planar embedding for each cluster, bottom up and level by level. This idea is adopted and similar in construction to the idea of Feng and Eades presented in their papers considering c-st-numbering and can be done in linear time concerning the number of vertices of G.

We call $G_{shrink}(\nu)$ the *cluster*-shrink graph of cluster ν of C. We construct $G_{shrink}(\nu)$ for each cluster ν as follows: First we assign the vertices of G to their clusters unique. A vertex v of

G is assign to a cluster ν if and only if ν is the cluster with the maximal length from the root of all clusters v belongs to. Hence, a vertex v of G is assigned to exactly one cluster, the parent cluster of the trivial cluster v corresponds to. Now we start from the leaves and traverse to the root cluster of the cluster tree T level by level. We construct auxiliary graphs for each level. For each non-trivial cluster ν we shrink its child cluster to a dummy vertex, combine every end vertex of every incident edge of ν that is not in ν into one dummy vertex and get the subgraph $G_{shrink}(\nu)$ (see Figure 4) . If $G_{shrink}(\nu)$ is biconnected, then there is no *cluster*-cut-vertex in $G(\nu)$ and we go to the next level. If there is one, then we add edges without connecting the dummy vertex to make $G_{shrink}(\nu)$ biconnected. As $G_{shrink}(\nu)$ consists of original edges before augmentation, we can also introduce the edges in the c-planar embedding of C very easily. We do this iteratively to the root cluster. Therefore, C is *cluster*-biconnected and we can triangulate it by triangulating G according Theorem 3. To triangulate the outer face we introduce a dummy vertex $_{dummy}$ and connect it with the end vertices of an edge $e = (v, w)$ so that the root cluster is the lowest common ancestor of v and w. The lowest common ancestor is the ancestor of the trivial clusters corresponding to v and w with the maximal length to the root. Further we introduce a dummy loop $e_{dummy} = (v_{dummy}, v_{dummy})$ as demonstrated in Figure 5. In the triangulation step, we do not connect v_{dummy}. Note that the added edges and dummy vertices are linear in size in the size of the original graph G. As $G - G(\nu)$ is non-connected for a cluster ν, we do not introduce double edges. As the triangulation do not introduce multiple edges, the resulting underlying graph G of the triangulated clustered graph C has no multiple edges. Hence G is triconnected and C still c-planar embedded.

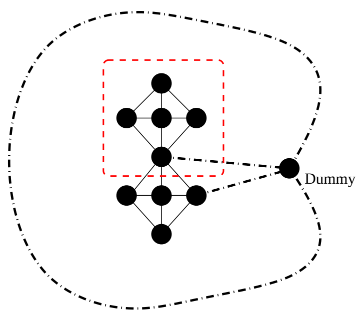

Fig. 5. Triangulating the outer face of a clustered graph C.

Theorem 4. *Let $C = (G, T)$ be a c-planar embedded c-connected clustered graph. C can be tested on cluster-biconnectivity in linear time concerning the number of vertices of G. If C is not cluster-biconnected, it can be augmented to a cluster-biconnected clustered graph without loosing the c-planar embedding in linear time.*

Theorem 5. *Let $C = (G, T)$ be a c-planar embedded c-connected clustered graph. C can be triangulated without loosing the c-planar embedding in linear time concerning the number of vertices of G. The triangulated underlying graph G does not contain multiple edges.*

Theorem 6. *For all faces f of the set of faces F of the c-planar embedded c-connected clustered graph $C = (G, T)$ exist a node $v \in f$, so that for all nodes $w \in f$ with $w \neq v$ and $w \notin adj(v)$ the following hold:*
The addition of the edge $e = (v, w)$ in G does not destroy the c-planar embedding of C.

A node v with this property is called an angle node.

Proof. Let f be a face of the c-planar embedded c-connected clustered graph $C = (G, T)$. Let ν be the cluster that has maximal length from the root cluster and contain all vertices belonging to f (therefore, ν is equal to the lowest common ancestor of all vertices belonging to f).

This face is contained in the graph $G_{shrink}(\nu)$. As $G_{shrink}(\nu)$ is biconnected, every vertex of the boundary of f have the property and is therefore an angle node. Otherwise, vertices of clusters that are not *cluster*-cut-vertices and on the boundary of f are angle nodes.

□

Algorithm 1: Triangulating a clustered graph C according to the given c-planar embedding.

Input: c-connected clustered graph $C = (G, T)$ with a c-planar embedding.
Result : triangulated c-connected clustered graph $C = (G, T)$ with a c-planar embedding.
Make C *cluster*-biconnected;
Triangulate G;

C-planar embedded clustered graphs have to be augmented to c-connected clustered graphs before triangulation, as we mentioned in the last section. A method was presented by Feng and Eades in the proof of Theorem 1 in [FCE95].

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.

[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+02a] C. Gutwenger, M. Jünger, S. Leipert, P. Mutzel, M. Percan, and R. Weiskircher. On c-planarity testing for non c-connected clustered graphs. Technical report, Institut für Informatik, Universität zu Köln, 2002. in preparation.

[GJL+02b] 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.

[HU91] T. Hagerup and C. Uhrig. Triangulating a planar graph, in library of efficient datatypes and algorithms (leda), software package. Technical report, Institut für Informatik, Saarbrücken, 1991.

[Kan93] G. Kant. Algorithms for drawing planar graphs. In *Ph D thesis*, 1993.

[Rea87] R.C. Read. A new method for drawing a graph given the cyclic order of the edges at each vertex. In *Congr. Numer.*, volume 56, pages 31–44, 1987.