How to Draw the Minimum Cuts of a Planar Graph ⋆

Ulrik Brandes

Fakultät für Mathematik & Informatik, Universität Passau
brandes@algo.fmi.uni-passau.de

Sabine Cornelsen

Dipartimento di Ingegneria Elettrica, Università dell'Aquila
cornelse@inf.uni-konstanz.de

Christian Fieß

SMA und Partner AG, Zürich
Christian@Fiess.com

Dorothea Wagner

Fakultät für Informatik, Universität Karlsruhe
dwagner@ira.uka.de

Abstract

We show how to utilize the cactus representation of all minimum cuts of a graph to visualize the minimum cuts of a planar graph in a planar drawing. In a first approach the cactus is transformed into a hierarchical clustering of the graph that contains complete information on all the minimum cuts. This approach is then extended to drawings in which the two vertex subsets of every minimum cut are separated by a simple closed curve. While both approaches work with any embedding-preserving drawing algorithm, we specifically discuss bend-minimum orthogonal drawings.

Key words: graph drawing, planar graphs, minimum cuts, cactus representation, clustered graphs

⋆ This work was partially supported by the DFG under grant BR 2158/1-1 and WA 654/13-1.

1 Introduction

The edge connectivity is a fundamental structural property of a graph. Dinitz et al. [5] discovered that the set of all minimum cuts of a connected graph G with positive edge weights has a tree-like structure. It can be represented by a cactus, i.e., by a connected graph in which every edge is contained in at most one cycle. Although the number of minimum cuts in a graph can be in $\Theta(n^2)$, the size of the cactus is linear in the number n of vertices of G. From the cactus representation, the bipartitions of the vertex set can easily be extracted, but it contains almost no information about the edges in G. We want to visualize a graph G together with the cactus representation of its minimum cuts in one drawing.

A simple closed curve divides the plane into two connected regions. A minimum cut divides the set of vertices of a graph into two connected subsets. Thus, it is natural to visualize a minimum cut in a drawing of a graph by a simple closed curve separating the two subsets. This leads to our general definition of drawings of sets of cuts in a graph. Each cut is represented by a simple closed curve that separates the corresponding two subsets of the vertex set. By requiring that only edges that connect the two subsets of the vertex set may cross the drawing of a cut, we guarantee that also these cut-edges are visualized. Finally, to avoid ambiguities, we require that each simple closed curve in the union of the drawing of all represented cuts also represents a cut of the given set. We show how to construct a planar drawing for the set of all minimum cuts of a weighted connected planar graph that meets these requirements.

The construction uses the model of hierarchically clustered graphs. This model was introduced by Feng et al. [13]. Since then algorithms for testing whether a hierarchically clustered graph has a planar drawing [13,4,18] for constructing planar drawings of hierarchically clustered graphs [8–12,22], for triangulating planar hierarchically clustered graphs [20] and for finding clusterings of graphs that respect its planarity [7] have been developed. In a drawing of a hierarchically clustered graph, a set of vertices of a graph is represented by a region that is bounded by a simple closed curve. The set of subsets of the vertex set that is represented simultaneously in this way has to have tree structure. In terms of cuts, this means that we can represent a set of pairwise non-crossing cuts as a hierarchically clustered graph. Graphs having no crossing minimum cuts are, for example, maximal planar graphs and chordal graphs.

If there are crossing cuts, the structure of the set of minimum cuts implies that they are represented at least implicitly in a drawing of the pairwise non-crossing cuts. We show, however, that the model of hierarchically clustered graphs can be extended to cactus-clustered graphs such that the goal of visualizing every minimum cut by a simple closed curve is achieved. This extension is mainly based on the fact that for two crossing minimum cuts, the four corner cuts are also minimum.

The contribution of this paper is as follows. Drawings of families of cuts are defined in Sect. 2. In Sect. 3 and 4, we provide some background on the cactus representation and on hierarchically clustered graphs, respectively. In Sect. 5, we show how to construct a hierarchically clustered graph from a cactus representation such that its c-planar drawing represents the set of pairwise non-crossing minimum cuts and we state our main theorem. Finally, our method for drawing planar graphs that are clustered according to all minimum cuts is presented in Sect. 6.

2 Drawings of Families of Cuts

Let $G = (V, E)$ be an undirected connected graph with n vertices. With $E(G)$ we denote the set E of edges of G and with $V(G)$ the set V of vertices of G. A cycle $c : v_1, \ldots, v_k$ is a sequence of $k \geq 3$ distinct vertices, such that $E(c) := \{\{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, v_1\}\} \subseteq E$. For a subset $E' \subseteq E$, we denote by $G - E'$ the graph $(V, E \setminus E')$.

A graph G together with a positive edge weight function $\omega : E \rightarrow \mathbb{R}^+$ is a *weighted graph*. For two subsets S and T of V, let $E(S,T) := \{\{v, w\}; v \in S \text{ and } w \in T\}$ be the set of edges between S and T, and let $\omega(S,T) := \sum_{e \in E(S,T)} \omega(e)$ be the total weight of the edges between the two subsets.

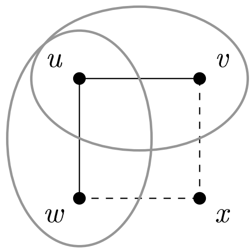
A *cut* is an unordered pair $\{S, \overline{S}\}$ where $\emptyset \subsetneq S \subsetneq V$ and $\overline{S} := V \setminus S$. A set S *induces* the cut $\{S, \overline{S}\}$. The *weight* of this cut is $\omega(S, \overline{S})$. With $\lambda := \min_{\emptyset \subsetneq S \subsetneq V} \omega(S, \overline{S})$ we denote the minimum of all these weights and a cut $\{S, \overline{S}\}$ of G satisfying $\omega(S, \overline{S}) = \lambda$ is called a *minimum cut*. With $\mathcal{M}(G)$ we denote the set of minimum cuts of G. By $G(S)$ we denote the subgraph of G induced by a set S. For an arbitrary cut $\{S, \overline{S}\}$, we do not require that $G(S)$ is connected. Note, however, that $G(S)$ is always connected if $\{S, \overline{S}\}$ is a minimum cut of a connected graph.

Let $\mathcal{C}$ be a set of cuts of G. A drawing of a cut should visualize both the partition of the vertex set into two parts and the edges with end-vertices in different parts. So, we define a *(planar) drawing* $\mathcal{D}$ of $(G, \mathcal{C})$ to be a map from elements of V, E, and $\mathcal{C}$ on subsets of $\mathbb{R}^2$. Each vertex v of G is represented as a distinct point $\mathcal{D}(v)$ and each edge $e = \{v, w\}$ as a simple curve $\mathcal{D}(e)$ between $\mathcal{D}(v)$ and $\mathcal{D}(w)$. (Drawings of edges do not intersect but in common end points.) Each cut $C = \{S, \overline{S}\} \in \mathcal{C}$ is represented by a simple closed curve $\mathcal{D}(C)$ such that

(1) $\mathcal{D}(S)$ and $\mathcal{D}(\overline{S})$ are in different connected regions of $\mathbb{R}^2 \setminus \mathcal{D}(C)$
(2) for every simple closed curve γ in $\bigcup_{C\in\mathcal{C}} \mathcal{D}(C)$
   (a) there is a cut $\{T, \overline{T}\} \in \mathcal{C}$ such that T and $\overline{T}$ are *separated* by γ, i.e. $\mathcal{D}(T)$ and $\mathcal{D}(\overline{T})$ are contained in different connected regions of $\mathbb{R}^2 \setminus \gamma$
   (b) and for every edge $e \in E$ it holds that $|\mathcal{D}(e) \cap \gamma| = \begin{cases} 1, \text{ if } |T \cap e| = 1 \\ 0, \text{ else} \end{cases}$

Condition 2 guarantees (a) that it is clear from a drawing of $(G, \mathcal{C})$ which cuts are in $\mathcal{C}$ and which are not and (b) that also the *cut-edges* of a cut $\{T, \overline{T}\} \in \mathcal{C}$, i.e. the edges in $E(T, \overline{T})$ are visualized – they are exactly the edges that cross the drawing of $\{T, \overline{T}\}$.

For example, let the dashed edges in the graph on the right have weight 1 and the solid edges weight 2. In this case $\lambda = 2$ and the picture on the right is a drawing for the set of all cuts of weight 2, 3, or 4. But there is no drawing for the set $\mathcal{C} = \{\ \{\{u, w\}, \{v, x\}\},\ \{\{u, v\}, \{w, x\}\}, \{\{u, v, w\}, \{x\}\}, \{\{x, v, u\}, \{w\}\}, \{\{u, x, w\}, \{v\}\}\ \}$ of cuts of weight 2 or 3: The union of the drawings of $\{\{u, w\}, \{v, x\}\}$ and $\{\{u, v\}, \{w, x\}\}$ would contain a simple closed curve γ with the property that $\{u\}$ and $\{v, w, x\}$ are contained in different connected regions of $\mathbb{R}^2 \setminus \gamma$. But $\{\{u\}, \{v, w, x\}\}$ is not in $\mathcal{C}$.



We will see, however, that the set of all minimum cuts of any weighted connected graph always allows a drawing. This fact is based on the almost tree-like structure – the so called cactus representation – of the set of all minimum cuts, which we briefly describe in the next section.

3 The Cactus of the Set of Minimum Cuts

Definition 1 *A* representation *for a set $\mathcal{C}$ of cuts of a graph G is a pair $(\mathcal{G}, \varphi)$ such that $\mathcal{G}$ is a weighted graph and $\varphi : V(G) \longrightarrow V(\mathcal{G})$ is a mapping such that $\mathcal{C} = \varphi^{-1}(\mathcal{M}(\mathcal{G})) := \{\{\varphi^{-1}(S), \varphi^{-1}(\overline{S})\}; \{S, \overline{S}\} \in \mathcal{M}(\mathcal{G})\}$. A node $\nu \in V(\mathcal{G})$ is called* empty *if $\varphi^{-1}(\nu) = \emptyset$.*

Definition 2 *Two cuts $\{S, \overline{S}\}$ and $\{T, \overline{T}\}$ are* crossing*, if none of the* corner sets *$S \cap T$, $S \cap \overline{T}$, $\overline{S} \cap T$, and $\overline{S} \cap \overline{T}$ is empty. A cut induced by a corner set is a* corner cut *and the cut induced by $S \triangle T := S \setminus T \cup T \setminus S$ is the* diagonal cut.

In the example in Sect. 2, the cuts induced by $\{u, v\}$ and $\{u, w\}$, respectively, cross. The corner cuts are the four cuts induced by $\{u\}$, $\{v\}$, $\{w\}$, and $\{x\}$.

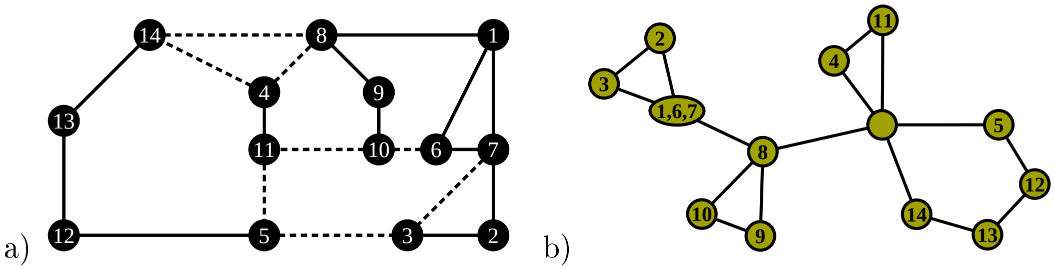


Fig. 1. a) A weighted connected graph and b) the cactus representation of its minimum cuts. In a), solid edges have weight 2 and dashed edges have weight 1. In b), φ is represented by the labels of the nodes.

The diagonal cut is the cut induced by $\{v, w\}$.

A cut is a *crossing cut* of a family $\mathcal{C}$ of cuts, if it crosses any cut in $\mathcal{C}$. If $\mathcal{C}$ contains no crossing cuts, $\mathcal{C}$ can be represented by a tree. Dinitz et al. [5] showed that the set of minimum cuts of an arbitrary weighted connected graph can be represented by a cactus where cycles correspond to sets of crossing cuts. More precisely:

Definition 3 (Cactus) *A* cactus *is a connected graph in which every edge belongs to at most one cycle. An edge that belongs to no cycle is called a* tree edge*. An edge that belongs to one cycle is called a* cycle edge.

In what follows, we assume that a weighted cactus is *uniform*, i.e. that all cycle edges have the same weight and that every tree edge has twice the weight of a cycle edge.

Theorem 4 ([5]) *The set $\mathcal{M}(G)$ of all minimum cuts of a weighted connected graph G has a representation $(\mathcal{G}, \varphi)$ such that $\mathcal{G}$ is a uniform cactus with $\mathcal{O}(n)$ nodes.*

Figure 1 shows an example of a weighted graph and its cactus. Dinitz and Nutov characterized all sets of cuts that can be represented by a cactus.

Theorem 5 ([6]) *A set $\mathcal{C}$ of cuts can be represented by a cactus if and only if for any two crossing cuts in $\mathcal{C}$*

- *the four corner cuts are in $\mathcal{C}$ and*
- *the diagonal cut is not in $\mathcal{C}$.*

If a cactus representation exists, there is always one with $O(n)$ nodes.

If a set $\mathcal{C}$ of cuts of G has a cactus representation $(\mathcal{G}, \varphi)$, a (planar) drawing of $(G, \mathcal{C})$ is called a *(planar) cactus-clustered drawing* of $(G, \mathcal{G}, \varphi)$.

In what follows, let $(\mathcal{G}, \varphi)$ be the cactus representation of a set of cuts of G.

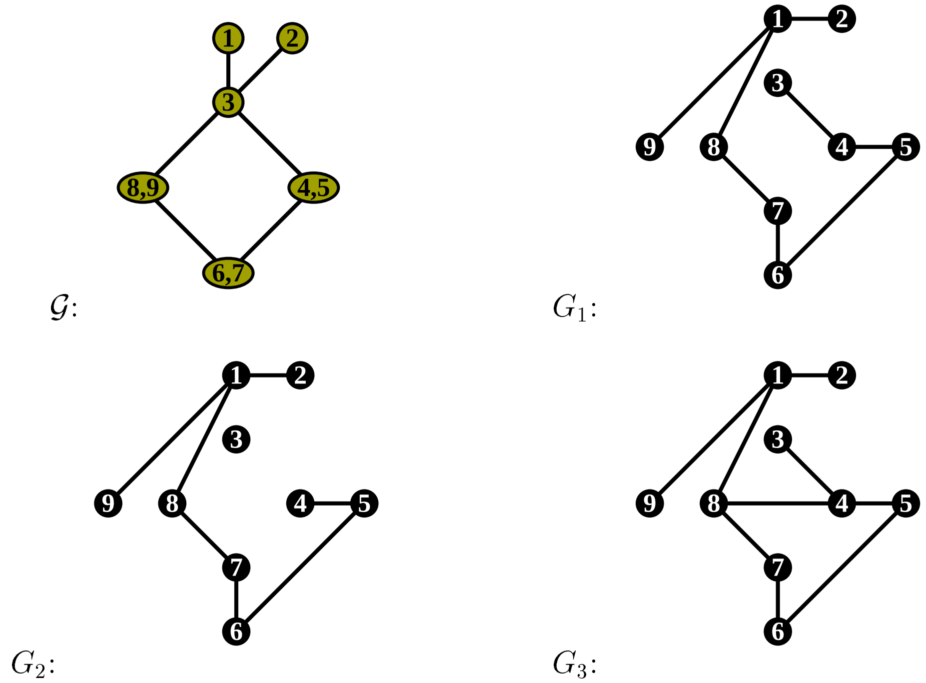


Fig. 2. The cycle in the cactus $\mathcal{G}$ corresponds to a cycle in the graph G_1, but not in the graphs G_2 and G_3. The mapping φ is represented by the labels of the nodes.

Note that there is a bijection between the set of minimum cuts of a cactus $\mathcal{G}$ and the set of tree edges and pairs of cycle edges belonging to the same cycle. Thus, we can also say that a cut in G is represented by a tree edge or by a pair of cycle edges of $\mathcal{G}$. The next definition is about relations between edges in G and cycles in $\mathcal{G}$.

Definition 6 *For a cycle $c : \nu_1, \ldots, \nu_k$ in $\mathcal{G}$ let $\mathcal{V}_i, i = 1, \ldots, k$, be the set of vertices in the connected component of $\mathcal{G} - E(c)$ that contains ν_i, and let $V_i := \varphi^{-1}(\mathcal{V}_i)$. We say that the cycle c of $\mathcal{G}$* corresponds to a cycle *of G if and only if*

$$E(V_i, V_j) \neq \emptyset \Longleftrightarrow i - j \equiv \pm 1 \bmod k.$$

An example can be found in Fig. 2. There is one cycle c in the cactus $\mathcal{G}$, with corresponding vertex sets $V_1 = \{1, 2, 3\}$, $V_2 = \{4, 5\}$, $V_3 = \{6, 7\}$, and $V_4 = \{8, 9\}$. This cycle corresponds to a cycle in G_1. Since $E(V_1, V_2) = \emptyset$ in the graph G_2, the cycle c does not correspond to a cycle in G_2. It does also not correspond to a cycle in G_3, because there $E(V_2, V_3) \neq \emptyset$.

For two crossing minimum cuts $\{S, \overline{S}\}$ and $\{T, \overline{T}\}$ of a graph with edge-connectivity λ it holds that (see, e.g., [5,14])

$$\omega(S \cap T, S \cap \overline{T}) = \omega(S \cap \overline{T}, \overline{S} \cap \overline{T}) = \omega(\overline{S} \cap \overline{T}, \overline{S} \cap T) = \omega(\overline{S} \cap T, S \cap T) = \lambda/2$$

and

$$\omega(\overline{S} \cap T, S \cap \overline{T}) = \omega(S \cap T, \overline{S} \cap \overline{T}) = 0.$$

Hence, there is the following useful property of the cactus of all minimum cuts.

Lemma 7 *If $\mathcal{G}$ is the cactus of all minimum cuts of a connected graph G, each cycle of $\mathcal{G}$ corresponds to a cycle of G.*

Fleischer [14] showed that the cactus of all minimum cuts of a weighted connected graph can be constructed in $\mathcal{O}(mn \log \frac{n^2}{m})$ time. For an unweighted graph, it can be computed in $\mathcal{O}(\lambda n^2)$ time [24]. Using the linear-time shortest-path algorithm of Henzinger et al. [19] for max-flow computations, the cactus of a weighted planar graph can be obtained in $\mathcal{O}(n^2)$ time with the construction described in [14].

4 Hierarchically Clustered Graphs

Feng et al. [13] introduced the hierarchically clustered graph model and characterized graphs that have a planar drawing with respect to the clustering. Such drawings are quite similar to cactus-clustered drawings in the special cases where the cactus is just a tree. In this section, we summarize definitions and results of [13] and [11] that we will use later.

A *hierarchically clustered graph* (G, T) consists of a graph $G = (V, E)$ and a rooted tree T such that the set of leaves of T is exactly V. Vertices of T are called *nodes*. Each node ν of T represents the *cluster* $V(\nu)$ of leaves in the subtree of T rooted at ν. T is called the *inclusion tree* of (G, T). An edge e of G is said to be *incident* to a cluster $V(\nu)$, if $|e \cap V(\nu)| = 1$.

A hierarchically clustered graph (G, T) is *connected*, if each cluster induces a connected subgraph of G.

A *c-planar drawing* $\mathcal{D}$ of a hierarchically clustered graph (G, T) consists of drawings of the underlying graph G and the inclusion tree T in the plane. Each vertex v of G is represented as a point $\mathcal{D}(v)$ and each edge $e = \{v, w\}$ as a simple curve $\mathcal{D}(e)$ between $\mathcal{D}(v)$ and $\mathcal{D}(w)$. The drawing of two edges may not intersect but in common end points. Each non-leaf node ν of T is drawn as a simple closed region $\mathcal{D}(\nu)$ bounded by a simple closed curve $\partial\mathcal{D}(\nu)$ such that

(1) $\mathcal{D}(\mu) \subseteq \mathcal{D}(\nu)$ for all descendants μ of ν.
(2) $\mathcal{D}(\mu) \cap \mathcal{D}(\nu) = \phi$ if μ is neither a descendent nor an ancestor of ν.

(3) for every edge e of G it holds that $|\mathcal{D}(e) \cap \partial\mathcal{D}(\nu)| = \begin{cases} 1, \text{ if } |V(\nu) \cap e| = 1 \\ 0, \text{ else} \end{cases}$.

Roughly speaking, T is drawn in the inclusion representation and edges of G may only cross cluster boundaries if necessary. A hierarchically clustered graph is *c-planar*, if it admits a c-planar drawing. In general, a hierachically clustered graph does not have to be c-planar if the underlying graph is planar. Feng et al. characterized connected c-planar hierarchically clustered graphs as follows.

Theorem 8 ([13]) *A connected hierarchically clustered graph $C = (G, T)$ is c-planar if and only if there exists a planar drawing of G, such that for each node ν of T all vertices of $V - V(\nu)$ are in the outer face of the drawing of $G(\nu)$.*

In an *OGRC (orthogonal grid rectangular cluster) drawing* of a hierarchically clustered graph (G, T), curve $\mathcal{D}(e)$ is a sequence of horizontal and vertical segments for every edge e of G and $\mathcal{D}(\nu)$ is an axis-parallel rectangle for every non-leaf node ν of T. Fig. 1 shows a connected hierarchically clustered graph with a c-planar OGRC-drawing.

Theorem 9 ([11]) *For a c-planar connected clustered graph with n vertices of degree at most 4, a c-planar OGRC-drawing with $\mathcal{O}(n^2)$ area and with at most 3 bends per edge can be constructed in $\mathcal{O}(n)$ time.*

In the following we will also use the notation OGRC-drawing for drawings of a family $\mathcal{C}$ of cuts of graph G in the corresponding sense, i.e., the drawing $\mathcal{D}(e)$ of an edge e of G is again a sequence of horizontal and vertical segments and every simple closed curve $\gamma \subseteq \bigcup_{C \in \mathcal{C}} \mathcal{D}(C)$ is an axis-parallel rectangle.

5 From Cactus Representations to Hierarchically Clustered Graphs

Both the cactus representation of a set of cuts of a graph and the inclusion tree of a hierarchically clustered graph represent structural information of a graph. Let $(\mathcal{G}, \varphi)$ be a linear sized cactus representation of a set $\mathcal{C}$ of cuts of a graph G with n vertices. As an intermediate step toward a cactus-clustered drawing, we transform the cactus representation into an inclusion tree such that a c-planar drawing of the corresponding hierarchically clustered graph yields a planar drawing of the set $\mathcal{C}_{nc}$ of pairwise non-crossing cuts of $\mathcal{C}$.

(1) For every cycle $c : \nu_1, \ldots, \nu_k$ in $\mathcal{G}$, delete all edges in c and add a new (empty) node ν_c and edges $\{\nu_i, \nu_c\}$, $i = 1, \ldots, k$.
(2) For every vertex v of G, add a new node ν_v and an edge $\{\varphi(v), \nu_v\}$.

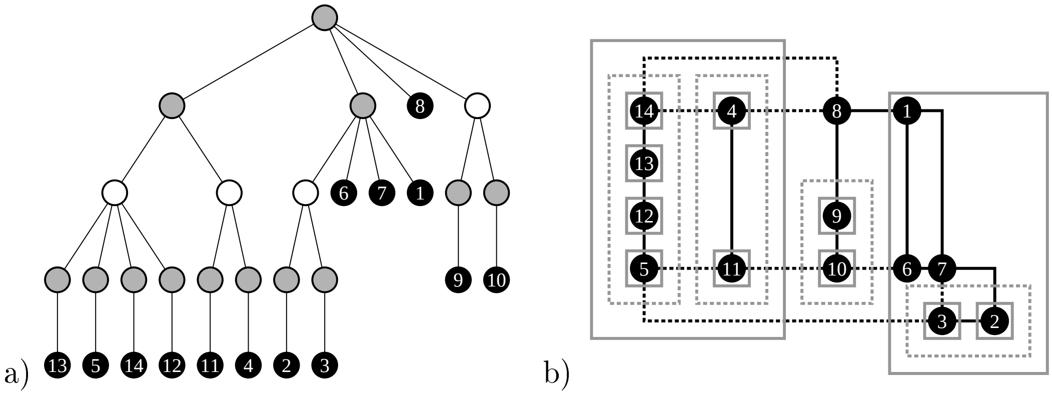


Fig. 3. a) White nodes in the inclusion tree $\mathcal{T}(G)$ of the graph G in Fig. 1 represent nodes that were added for a cycle in $\mathcal{G}$. b) The corresponding cluster boundaries are drawn as dashed grey rectangles in the c-planar OGRC-drawing of $(G, \mathcal{T}(G))$.

(3) Find a suitable root r.

We call the thus constructed rooted tree $\mathcal{T} = \mathcal{T}(\mathcal{G}, \varphi, r)$. In the special case where $(\mathcal{G}, \varphi)$ is the cactus of all minimum cuts we refer to $\mathcal{T}$ by $\mathcal{T}(G)$. Note that $(G, \mathcal{T})$ is now a hierarchically clustered graph. Also note that there might be nodes of degree two in $\mathcal{T}$, thus some clusters might be represented twice in $(G, \mathcal{T})$, but the number of nodes in $\mathcal{T}$ is still in $\mathcal{O}(n)$: By Theorem 5, we have $|V(\mathcal{G})| \in \mathcal{O}(n)$. In Step 1, we add a new node for every cycle in $\mathcal{G}$ and in Step 2, we add n new nodes. Thus $|V(\mathcal{T})|$ is in $\mathcal{O}(n)$, as well.

Figure 3 shows the inclusion tree $\mathcal{T}(G)$ of the graph G from Fig. 1. There are several options for choosing a root. We have chosen the root such that $|V(\nu)| \leq |\overline{V(\nu)}|$ for every inner node ν of $\mathcal{T}(G)$. This has the advantage that the most balanced minimum cut $\{S, \overline{S}\}$, i.e. the cut such that $||S| - |\overline{S}||$ is minimal, is seen on the top level. Another possibility is to take the center of the tree, i.e. to minimize the height. In either case, the root can be computed in linear time.

From the construction of $\mathcal{T}$ it follows immediately that

$$\mathcal{C}_{nc} = \{\{V(\nu), \overline{V(\nu)}\};\ \nu \neq r \text{ is a non-leaf node of } \mathcal{T}\}. \tag{1}$$

Thus, if $\mathcal{D}$ is a c-planar drawing of $(G, \mathcal{T})$ and $\mathcal{D}'$ is defined by $\mathcal{D}'(v) = \mathcal{D}(v)$, $\mathcal{D}'(e) = \mathcal{D}(e)$, and $\mathcal{D}'(\{V(\nu), \overline{V(\nu)}\}) = \partial\mathcal{D}(\nu)$ for vertices $v \in V$, edges $e \in E$, and non-leaf nodes $\nu \neq r$ of $\mathcal{T}$ then $\mathcal{D}'$ is a planar drawing of $(G, \mathcal{C}_{nc})$.

In the rest of the paper we will show the following theorem and its application to the set of all minimum cuts of a weighted planar connected undirected graph.

Theorem 10 (Main Theorem)
Let $(\mathcal{G}, \varphi)$ be a cactus representation of a set of cuts of G such that each cycle of

$\mathcal{G}$ corresponds to a cycle in G. Then there is a planar cactus-clustered drawing of $(G, \mathcal{G}, \varphi)$ if and only if $(\mathcal{G}, \mathcal{T}(\mathcal{G}, \varphi, r))$ is c-planar for a suitable choice of the root r.

If $(\mathcal{G}, \mathcal{T}(\mathcal{G}, \varphi, r))$ is a c-planar connected hierarchically clustered graph and h is the height of the inclusion tree $\mathcal{T}$, a bend-minimum planar cactus-clustered OGRC-drawing of $(G, \mathcal{G}, \varphi)$ can be constructed in $\mathcal{O}((n \cdot h)^{7/4}\sqrt{\log n})$ time.

To show that we can apply the main theorem to the cactus of all minimum cuts, we show how to construct a c-planar drawing of the hierarchically clustered graph $(G, \mathcal{T}(G))$. The next lemma guarantees that we can fix an arbitrary embedding of G and either the root of $\mathcal{T}(G)$ or the outer face of G and add cluster boundaries. See also [3,20] for related results.

Lemma 11 *Every planar drawing of a weighted connected planar graph G can be extended to a c-planar drawing of the connected hierarchically clustered graph $(G, \mathcal{T}(G))$.*

PROOF. If $\{S, \overline{S}\}$ is a minimum cut in a weighted connected planar graph, then the following holds.

(1) $G(S)$ and $G(\overline{S})$ are both connected.
(2) For any embedding of G, the dual edges of $E(S, \overline{S})$ induce a cycle.

These two facts guarantee that for every planar embedding of the weighted connected graph G, provided either the root of $\mathcal{T}(G)$ or the outer face of G is chosen in such a way that for each non-leaf node ν of $\mathcal{T}(G)$, cluster $V(\nu)$ is inside the dual cycle of $E(V(\nu), \overline{V(\nu)})$, the hierarchically clustered graph $(G, \mathcal{T}(G))$ fulfills the preconditions of Theorem 8 and thus has a c-planar drawing. $\square$

Now, by Theorem 4 and Lemma 7, we have the following corollary of the main theorem.

Corollary 12 *There is a planar drawing of $(G, \mathcal{M}(G))$.*

It remains to show the main theorem. So let $(\mathcal{G}, \varphi)$ be a cactus representation of a set $\mathcal{C}$ of cuts of G such that no edge of G crosses a cycle of $\mathcal{G}$. Suppose first that there is a planar cactus-clustered drawing $\mathcal{D}$ of $(G, \mathcal{G}, \varphi)$. Choose the root r of $\mathcal{T}(\mathcal{G}, \varphi, r)$ in such a way that for any node ν of $\mathcal{T}(\mathcal{G}, \varphi, r)$ the cluster $V(\nu)$ is enclosed by $\mathcal{D}(V(\nu), \overline{V(\nu)})$. We construct a c-planar drawing $\mathcal{D}'$ for $(G, \mathcal{T}(\mathcal{G}, \varphi, r))$ by extending drawing $\mathcal{D}$ of the underlying graph G, i.e. $\mathcal{D}'(v) = \mathcal{D}(v)$ and $\mathcal{D}'(e) = \mathcal{D}(e)$ for every vertex $v \in V$ and every edge $e \in E$. To guarantee Property 2 for the cluster boundaries, we construct the following

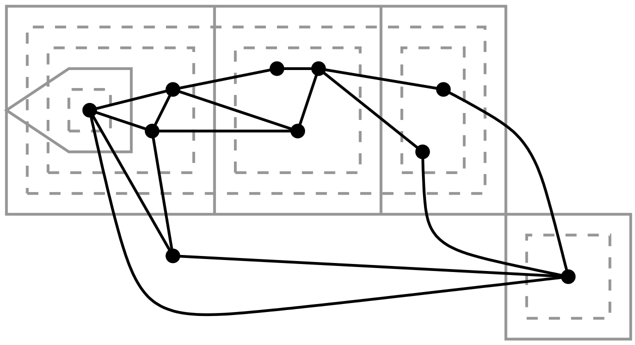


Fig. 4. Illustration of the set $\mathcal{S}$. Drawings of cuts are solid grey curves and elements of $\mathcal{S}$ are indicated as dashed grey curves.

set $\mathcal{S}$. Consider the simple closed curves in $\bigcup_{C\in\mathcal{C}}\mathcal{D}(C)$ ordered such that γ_1 is before γ_2 if γ_1 is completely contained in the simple closed region bounded by γ_2. For every simple closed curve $\gamma \subseteq \bigcup_{C\in\mathcal{C}}\mathcal{D}(C)$ that is completely contained in the closure of a connected region of $\mathbb{R}^2 \setminus \bigcup_{C\in\mathcal{C}}\mathcal{D}(C)$, set $\mathcal{S}$ contains a simple closed curve γ' completely contained in the interior of the region bounded by γ such that

- γ and γ' separate the same vertex sets,
- intersect the same edges in the same order and in the same number of times, and
- such that γ' does not intersect any other curve in $\mathcal{S}$.

By a consequence of the Schönflies theorem [1] (see e.g. [23, p. 76]), $\mathcal{S}$ is well-defined. Figure 4 illustrates the set $\mathcal{S}$.

Lemma 13 $\{\{S,\overline{S}\};$ *S and $\overline{S}$ are separated by a curve of $\mathcal{S}\}$ equals the set $\mathcal{C}_{nc}$ of all non-crossing cuts in $\mathcal{C}$.*

PROOF. If S and $\overline{S}$ are separated by a curve $\gamma \subseteq \bigcup_{C\in\mathcal{C}}\mathcal{D}(C)$ but not by a curve in $\mathcal{S}$, then $\bigcup_{C\in\mathcal{C}}\mathcal{D}(C)$ contains two paths – one that lies completely inside γ and one that lies completely outside γ and each have both end points on γ. These two paths together with non-crossing connections of there end points on γ form a simple closed cycle in $\bigcup_{C\in\mathcal{C}}\mathcal{D}(C)$. The corresponding cut crosses $\{S,\overline{S}\}$. Hence $\{S,\overline{S}\} \notin \mathcal{C}_{nc}$.

If, on the other hand, two cuts $\{S,\overline{S}\},\{T,\overline{T}\} \in \mathcal{C}$ cross, no simple cycle that separates S from $\overline{S}$ can be contained in the closure of a connected component of $\mathbb{R}^2 \setminus \bigcup_{C\in\mathcal{C}}\mathcal{D}(C)$. □

[1] The Schönflies theorem says that a homeomorphism of a simple closed curve in the plane onto a circle in the plane can be extended to a homeomorphism of the entire plane.

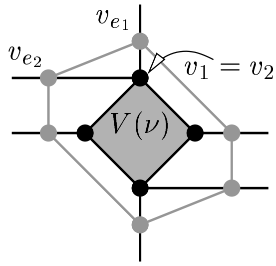
Thus, by Equation 1, the curves in $\mathcal{S}$ together with a simple closed curve that contains the whole graph contain the cluster boundaries for a c-planar drawing of $(G, \mathcal{T}(\mathcal{G}, \varphi, r))$.

Now, suppose that $(G, \mathcal{T}(\mathcal{G}, \varphi, r))$ is c-planar for some root r. We want to construct a planar cactus clustered drawing of $(G, \mathcal{G}, \varphi)$ via a c-planar drawing of $(G, \mathcal{T}(\mathcal{G}, \varphi, r))$. Thus, in the next section, we first introduce a method for drawing hierarchically clustered c-planar graphs. We will then show how to extend this method to planar cactus clustered graphs.

6 The Drawing

6.1 C-Planar Drawings of Hierarchically Clustered Graphs

As mentioned in Sect. 4, Eades et al. [11] introduced a method for drawing hierarchically clustered planar graphs orthogonally with rectangularly shaped cluster boundaries. In this method, the undirected graph is made directed and edges are allowed to cross cluster boundaries only at the top or bottom of the boundary rectangle. It might therefore introduce unnecessary bends into the drawing. We propose a different way of drawing a connected c-planar graph (G, T). We add edges and vertices to G such that the newly constructed graph G' remains planar and each cluster boundary corresponds to a cycle in G'. Now any embedding preserving algorithm can be applied to draw graph G' and thus to obtain a c-planar drawing of (G, T). In case G has maximum degree 4, using the model of Tamassia [25] with some additional constraints on the flow, this leads to a c-planar OGRC-drawing with the minimum number of bends. This extension of Tamassia's model to hierarchically clustered graphs was independently described in [22] and is now part of the AGD library [17]. The approach is easily extended to graphs with arbitrary degree by using near orthogonal drawings [26,15,21]. Dynamical approaches for drawing graphs using a flow model can be found, e.g., in [2,1].



Let (G, T) be a hierarchically clustered c-planar graph with an embedding in the plane that fulfills the conditions of Theorem 8. Recall, that for $T = \mathcal{T}(G)$, by Lemma 11, every embedding of a planar graph is suitable. For every non-leaf node of the inclusion tree, we add a cycle of new edges and new vertices to G in the following way: Proceeding from the leaves to the root of T, for every non-leaf node ν of T let $e_1, \ldots, e_k$ be the edges incident to cluster $V(\nu)$ in their cyclic order around $V(\nu)$. Let $e_{k+1} = e_1$ and $e_i = \{v_i, w_i\}$, $i = 1, \ldots, k+1$. For

$i = 1, \ldots, k$, we *split* edge e_i, i.e. we add a vertex v_{e_i} to $V(G)$ and replace edge e_i by edges $\{v_i, v_{e_i}\}$ and $\{w_i, v_{e_i}\}$. Finally, we add edges $\{v_{e_i}, v_{e_{i+1}}\}$. These k edges are called *boundary edges* of ν. They form a cycle, called the *boundary cycle* of ν, that model the boundary of $\mathcal{D}(\nu)$.
A special case occurs, if there are only one or two edges incident to a cluster. In that case, two or one additional vertices are inserted in this cycle to avoid loops and multiple edges. The added edges are also called boundary edges. Let the resulting graph be G'. Let $n = |V(G)|$ and h be the height of the inclusion tree T.

Lemma 14 $|V(G')| \in \mathcal{O}(n \cdot h)$

PROOF. Let $e = \{u, v\} \in E(G)$ and let k be the number of vertices on the path in T between u and v. Then $k - 3 \leq 2h$ vertices are inserted into e. Thus, $|V(G')| \leq n + 2|E(G)|h \in \mathcal{O}(n \cdot h)$. □

Note that in case $T = \mathcal{T}(G)$ and $\omega(e) \geq 1$ for every edge $e \in E(G)$ it is also true, that $|V(G')| \in \mathcal{O}(\lambda \cdot n)$: Every cluster is incident to at most λ edges and the number of clusters is in $\mathcal{O}(n)$. For unweighted planar graphs we have $\lambda \leq 5$ and this implies $|V(G')| \in \mathcal{O}(n)$.

Lemma 15 *If (G, T) is a connected hierarchically clustered graph, G' can be constructed in $\mathcal{O}(|V(G')|)$ time.*

PROOF. Proceeding for each edge $\{u, v\} \in E(G)$ along the path in T between u and v, splitting the edges can be done in $\mathcal{O}(|E(G)| + |\text{added vertices}|)$ $= \mathcal{O}(|V(G')|)$.

From the leave to the root of T, add the boundary edges along the outer face of each cluster. Doing this, every edge can be touched at most twice. Thus, inserting the boundary edges is in $\mathcal{O}(|E(G')|) = \mathcal{O}(|V(G')|)$. □

In the flow network for an orthogonal or near orthogonal drawing of G', we restrict the flow over a boundary edge to be zero, if it goes from outside the corresponding boundary cycle into it. This guarantees that the boundary cycles are rectangularly shaped in any resulting orthogonal drawing. Theorem 9 guarantees that there is a feasible flow for the restricted flow network. The resulting drawing is a bend minimum c-planar OGRC-drawing. Moreover, all inserted vertices have degree 4 and split edges alternate with boundary edges. Thus, the corresponding original edges in G have no bends at cluster boundaries. The restriction on the flow is necessary. Even in the case of unweighted graphs with the root chosen in such a way, that $|V(\nu)| \leq |\overline{V(\nu)}|$, there are

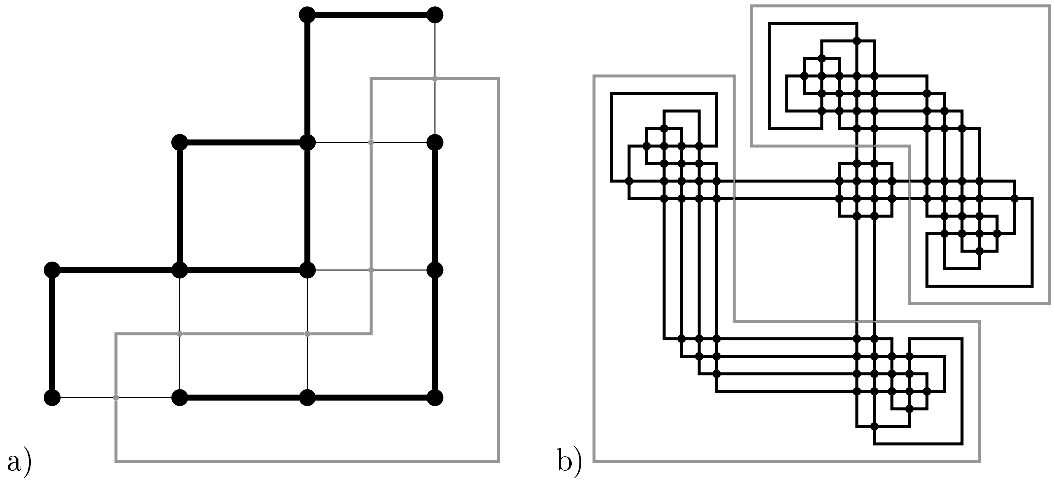


Fig. 5. Drawings of the non-trivial minimum cuts of a) a weighted and b) an unweighted graph without rectangularity restriction on the cluster shape. Grey edges are boundary edges, thin edges have weight 1 and thick edges have weight 6.

examples of planar graphs G such that the bend minimum solution without restriction of the clustered graph $(G, \mathcal{T}(G))$ have non-rectangularly shaped cluster boundaries. See for example Fig. 5.

Lemma 16 *The area of the thus constructed bend-minimum c-planar OGRC-drawing of* (G, T) *is* $\mathcal{O}(n^2)$.

PROOF. There are $\mathcal{O}(n)$ clusters and each cluster boundary requires two horizontal and two vertical lines.

Those edges in G' that are not boundary edges correspond to $\mathcal{O}(n)$ original edges in G. As the constructed drawing is bend-minimum, by Theorem 9, there are at most $3 \cdot |E(G)|$ bends on those edges. Thus, the non-boundary edges require at most $4 \cdot |E(G)| \in \mathcal{O}(n)$ horizontal and vertical lines. □

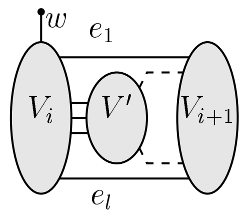
Figure 3 shows a bend-minimum c-planar OGRC-drawing of the hierarchically clustered graph $(G, \mathcal{T}(G))$ where G is the graph in Fig. 1.

6.2 Planar Cactus-Clustered Drawings

In this subsection $(\mathcal{G}, \varphi)$ continues to be a linear sized cactus representation of a set of cuts of the graph G and $\mathcal{T} = \mathcal{T}(\mathcal{G}, \varphi, r)$. We show how we can transform a c-planar drawing of $(G, \mathcal{T})$ into a cactus-clustered drawing of $(G, \mathcal{G}, \varphi)$. We achieve this, roughly speaking, by merging the cluster boundaries corresponding to pairs of incident nodes on a cycle in the cactus. In Step 1, we replace each cycle of the cactus by a star. Thus, the information about the cyclic order of the edges in a cycle of $\mathcal{G}$ is not preserved in $\mathcal{T}$. However, this order can be reconstructed from a c-planar drawing of $(G, \mathcal{T})$ by the fact that cycles in $\mathcal{G}$ correspond to cycles in G.

Let $c : \nu_1, \ldots, \nu_k$ be a cycle in $\mathcal{G}$ and let V_i be defined as in Definition 6. Note that for each i either V_i or $\overline{V_i}$ is a cluster of $(G, \mathcal{T})$. More precisely, let ν_c be the node that was added for c in Step 1. If ν_i is a descendent of ν_c then $V_i = V(\nu_i)$. If ν_i is a ancestor of ν_c then $\overline{V_i} = V(\nu_c)$. In what follows, suppose without loss of generality, that the root r of $\mathcal{T}$ is not node ν_c and that ν_k is an ancestor of ν_c. To associate every node in $\mathcal{T}(G)$ with at most one cycle in $\mathcal{G}$, we will associate ν_c with c instead of its ancestor ν_k and, for an easier notation, we will denote ν_c also by ν_k. For convenience, we will refer to the indices of nodes in c as if taken modulo k. Consider the sequences d_i of edges in $E(V_i, \overline{V_i})$ in their cyclic order around $V(\nu_i)$. Since each cycle of $\mathcal{G}$ correspond to a cycle of $\mathcal{G}$, we already know that $E(V_i, \overline{V_i}) = E(V_i, V_{i-1}) \cup E(V_i, V_{i+1})$. The next lemma guarantees that these two sets form intervals in d_i.

Lemma 17 *The set of edges $E(V_i, \overline{V_i}) \cap E(V_{i+1}, \overline{V_{i+1}})$ is consecutive in d_i and d_{i+1}.*



PROOF. If not, let $e_1, \ldots, e_l$ be the subsequence of d_i such that $e_1, e_l \in E(V_{i+1}, \overline{V_{i+1}})$, $e_2, \ldots, e_{l-1} \notin E(V_{i+1}, \overline{V_{i+1}})$. Let $e \in E(V_i, \overline{V_i}) \setminus (E(V_{i+1}, \overline{V_{i+1}}) \cup \{e_2, \ldots, e_{l-1}\})$ be another edge in d_i and let $e = \{v, w\}$ such that $w \notin V(\nu_i)$. For $j = i, i+1$ let p_j be a path on the cluster boundary of U_j from e_1 to e_l. Let c' be the simple closed cycle that is induced by edge e_1, path p_i, edge e_l and path p_{i+1}. Without loss of generality we can assume that edges $e_2, \ldots, e_{l-1}$ are inside or intersect cycle c'. Let $V' \subseteq V(G)$ be the set of vertices that are incident to $e_2, \ldots, e_{l-1}$ and that are not in V_i. Then, since no edge of G crosses a cycle of $\mathcal{G}$, $w \in V_{i-1}$ and $V' \subseteq V_{i-1}$. Thus, V_{i-1} cannot be bounded by a simple closed curve that intersects neither e_1 nor e_ℓ nor the cluster boundary of $V(\nu_i)$ or $V(\nu_{i+1})$. □

Let v_e^i be the vertex that was inserted into an edge e for the boundary cycle of ν_i. It follows from the previous lemma that the boundary cycle of ν_i is divided into the following four parts: two paths p_i^+ and p_i^- that are induced by the vertex sets $\{v_e^i;\ e \in E(V_i, V_{i\pm1})\}$ and the two remaining edges. The next lemma guarantees that paths p_i^+ and p_{i+1}^- are adjacent.

Lemma 18 *If $e \in E(V_i, V_{i+1})$ then $\{v_e^i, v_e^{i+1}\} \in E(G')$.*

PROOF. Suppose there was another vertex inserted into e between v_e^i and v_e^{i+1} for the cluster boundary of the cluster represented by a node ν. Then

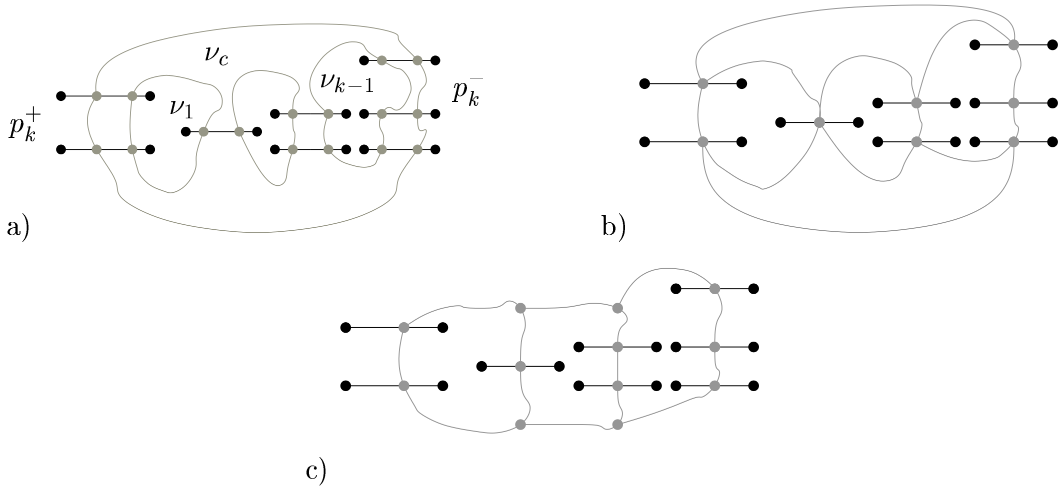


Fig. 6. Constructing a planar cactus-clustered drawing from a c-planar drawing.

either ν_i and ν_{i+1} are both descendants of ν_c or one of them, say $\nu_{i+1} = \nu_c$. In the first case, $V(\nu)$ contains exactly one of $V(\nu_i)$ and $V(\nu_{i+1})$, say $V(\nu_i)$. Thus ν is an ancestor of ν_i but not of ν_{i+1}. In the second case, $V_i \subseteq V(\nu) \subseteq V(\nu_c)$. Both cases are impossible, since ν_i is adjacent to ν_c. $\square$

Thus, we have the situation indicated in Fig. 6a: a path of adjacent clusters $V(\nu_1), \ldots, V(\nu_{k-1})$ surrounded by the boundary cycle of ν_c. Now, for each $i = 1 \ldots k$ and for each edge $e \in E(V_i, V_{i+1})$, we can merge vertices v_e^i and v_e^{i+1} without loosing planarity. The result is shown in Fig. 6b. Now, for each $i = 1 \ldots k$, paths p_i^+ and p_{i+1}^- are united into one path p_{i+1}. For $i = 2, \ldots, k-1$ we add two vertices to p_i, one on each end of p_i. We will call these new vertices *cycle-path end-vertices*. We replace an incidence of a remaining edge of the cluster boundaries of ν_i and ν_{i-1} to p_i by the corresponding new cycle-path end-vertex of p_i. Finally, the remaining edges of the boundary cycle of ν_c are deleted. The result is shown in Fig. 6c. The simple closed cycles contained in the thus modified cluster boundaries of $\nu_1, \ldots, \nu_k$ separate exactly the sets $\bigcup_{\ell=i}^{j} V_\ell$, $1 \leq i \leq j < k$, from their complement. These are exactly the sets that are modeled by c in $\mathcal{G}$.

Let G'' be the graph in which the above described construction is done for every cycle in $\mathcal{G}$. As the number of cycles in $\mathcal{G}$ is in $\mathcal{O}(n)$, we add $\mathcal{O}(n)$ vertices to G'. Thus, $|V(G'')| \in \mathcal{O}(|V(G')|)$ and G'' can be constructed in $\mathcal{O}(|V(G')|$ time.

As in the previous subsection, we can now apply any embedding preserving algorithm to draw graph G'' and thus to get a cactus-clustered drawing of $(G, \mathcal{G}, \varphi)$. To achieve a bend-minimum planar cactus-clustered OGRC-drawing, we can apply the flow model of Tamassia [25] to G'' with similar constraints on the flow as in the previous subsection. Again, we restrict the flow over a boundary edge to be zero, if it goes from outside the corresponding boundary cycle into it. The flow from a cycle-path end-vertex into a cluster is restricted to 1. This has the effect that every simple cycle in G'' that consists of boundary-edges is drawn as a rectangle.

Lemma 19 *There is a feasible flow for the restricted flow network.*

PROOF. Let c be a cycle of $\mathcal{G}$ and the notations as above. We modify an orthogonal drawing of G' in such a way that

(1) all edges of $E(V_i, V_{i+1})$ leave the cluster boundary of ν_i on the same side and all edges of $E(V_i, V_{i-1})$ on the opposite side.
(2) for an edge $e \in E(V_i, V_{i+1})$ edge $\{v_e^i, v_e^{i+1}\}$ is a straight line.

These two properties are achieved by pushing flow along cycles in the flow network as indicated in Fig. 7. In the first step (Fig. 7a), the bends in the boundary cycles are moved along the boundary cycles to the desired place. Now, for each $e \in E(V_i, V_{i+1})$ the number of bends in $\{v_e^i, v_e^{i+1}\}$ is the same. In the second step (Fig. 7b), these bends are all moved to the edges $\{v_e^k, v_e^1\}$. Since the edges in $E(V_k, V_1)$ and $E(V_k, V_{k-1})$ leave the cluster boundary of ν_c in opposite directions, in the end the edges $\{v_e^k, v_e^1\}$ are also straight. Doing this for every cycle, results in such a drawing that merging corresponding cluster-sides automatically results in a – not necessarily bend-minimum – planar cactus-clustered OGRC-drawing. This drawing corresponds to a flow in the restricted flow network. □

An example using the construction of a bend-minimum planar cactus-clustered OGRC-drawing is shown in Fig. 7d. If $(G, \mathcal{T})$ is connected, the running time of the algorithm is as follows:

- Constructing the inclusion tree $\mathcal{T}$ of height h from the cactus is in $\mathcal{O}(n)$.
- Constructing G' and G'' from $(G, \mathcal{T})$ is in $\mathcal{O}(n \cdot h)$.
- Constructing the orthogonal drawing of G'' with $N := |V(G')| \in \mathcal{O}(n \cdot h)$ vertices is in $\mathcal{O}(N^{7/4}\sqrt{\log N})$ [16].

We can finally summarize that the running time is dominated by the orthogonal drawing and is in $\mathcal{O}((n \cdot h)^{7/4}\sqrt{\log n})$ time.

7 Conclusion and Future Work

We outlined a method for representing the minimum cuts of a weighted planar graph in a planar drawing of the graph. Utilizing the cactus representation, the set of all mutually non-crossing minimum cuts can be shown in a

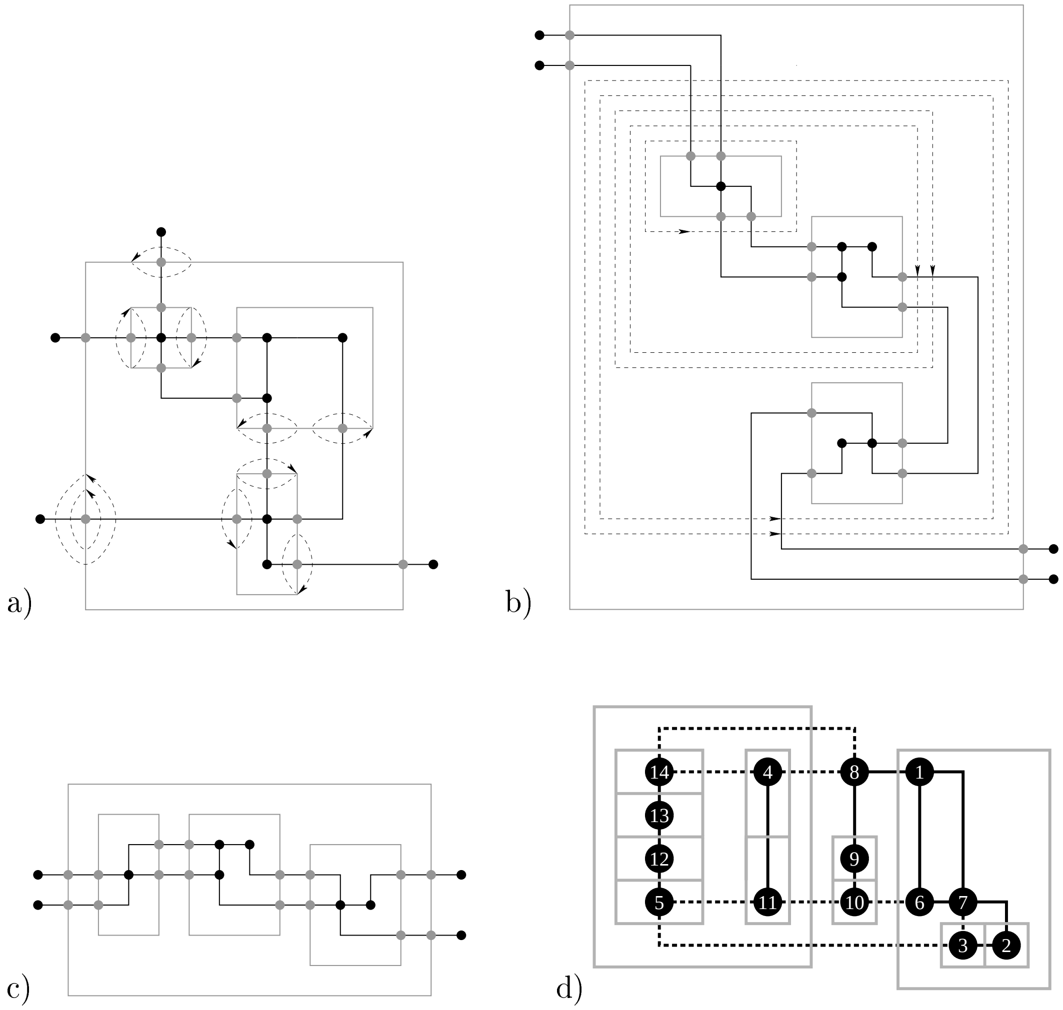


Fig. 7. a)-c)Bends in an orthogonal drawing of G' are moved along the dashed cycles. d) A cactus-clustered drawing of the graph in Fig. 1

c-planar drawing of a hierarchical clustering of the graph. This approach was then extended to cactus-clustered drawings that visualize all minimum cuts by simple closed curves. Both approaches have been demonstrated to work for bend-minimum orthogonal drawings, but can be used with any drawing algorithm that preserves the embedding of cluster boundaries.

Moreover, our method applies to any set $\mathcal{C}$ of, not necessarily minimum, cuts of a planar graph G that has a cactus representation $(\mathcal{G}, \varphi)$ and the additional property that

each cycle of $\mathcal{G}$ corresponds to a cycle of G.

If $\mathcal{T}$ is the inclusion tree constructed from $\mathcal{G}$ as described in Sect. 5, it holds that $(G, \mathcal{G}, \varphi)$ has a planar cactus-clustered drawing if and only if $(G, \mathcal{T})$ is c-planar for a suitable choice of the root of $\mathcal{T}$.

Eades et al. give a linear-time algorithm that constructs a c-planar straight-line hierarchically clustered drawing in which the clusters are drawn as trapezoids [11]. It would be interesting to know whether there exist cactus-clustered drawings of this kind.

References

[1] U. Brandes, M. Eiglsperger, M. Kaufmann, and D. Wagner. Sketch-driven orthogonal graph drawing. In M. T. Goodrich and S. G. Kobourov, editors, *Proceedings of the 10th International Symposium on Graph Drawing (GD 2002)*, volume 2528 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2002.

[2] S. Bridgeman, J. Fanto, A. Garg, R. Tamassia, and L. Vismara. Interactive giotto: An algorithm for interactive orthogonal graph drawing. In G. Di Battista, editor, *Proceedings of the 5th International Symposium on Graph Drawing (GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 303–308. Springer, 1998.

[3] S. Cornelsen and D. Wagner. Completely connected clustered graphs. To appear in *Proceedings of the 29th International Workshop on Graph Theoretic Concepts in Computer Science (WG 2003).*

[4] E. Dahlhaus. A linear time algorithm to recognize clustered planar graphs and its parallelization. In C. L. Lucchesi and A. V. Moura, editors, *Proceedings of the 3rd Latin American Symposium on Theoretical Informatics (LATIN '98)*, volume 1380 of *Lecture Notes in Computer Science*, pages 239–248. Springer, 1998.

[5] Y. Dinitz, A. V. Karzanov, and M. Lomonosov. On the structure of a family of minimal weighted cuts in a graph. In A. Fridman, editor, *Studies in Discrete Optimization*, pages 290–306. Nauka, 1976. (in Russian).

[6] Y. Dinitz and Z. Nutov. A 2-level cactus model for the system of minimum and minimum+1 edge–cuts in a graph and its incremental maintenance. In *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing (STOC '95)*, pages 509–518. ACM, The Association for Computing Machinery, 1995.

[7] C. Duncan, M. T. Goodrich, and S. G. Kobourov. Planarity-preserving clustering and embedding for large planar graphs. *Computational Geometry*, 24(2):95–114, 2003.

[8] P. Eades, R. F. Cohen, and Q. Feng. How to draw a planar clustered graph. In D.-Z. Du and M. Li, editors, *Proceedings of the 1st Annual International Conference on Coputing and Combinatorics (COCOON '95)*, volume 959 of *Lecture Notes in Computer Science*, pages 21–30. Springer, 1995.

[9] P. Eades and Q. Feng. Multilevel visualization of clustered graphs. In S. C. North, editor, *Proceedings of the 4th International Symposium on Graph Drawing (GD '96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 101–112. Springer, 1996.

[10] P. Eades, Q. Feng, and X. Lin. Straight-line drawing algorithms for hierarchical graphs and clustered graphs. In S. C. North, editor, *Proceedings of the 4th International Symposium on Graph Drawing (GD '96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 146–157. Springer, 1996.

[11] P. Eades, Q. Feng, and H. Nagamochi. Drawing clustered graphs on an orthogonal grid. *Journal on Graph Algorithms and Applications*, 3(4):3–29, 1999.

[12] P. Eades, H. Nagamochi, and Q. Feng. Straight-line drawing algorithms for hierarchical graphs and clustered graphs. Technical Report 98-03, Department of Computer Science and Software Engineering, University of Newcastle, Australia, 1998. Available at `ftp://ftp.cs.newcastle.edu.au/pub/techreports/tr98-03.ps.Z`.

[13] Q. Feng, R. F. Cohen, and P. Eades. Planarity for clustered graphs. In P. Spirakis, editor, *Proceedings of the 3rd European Symposium on Algorithms (ESA '95)*, volume 979 of *Lecture Notes in Computer Science*, pages 213–226. Springer, 1995.

[14] L. Fleischer. Building chain and cactus representations of all minimum cuts from Hao–Orlin in the same asymptotic run time. *Journal of Algorithms*, 33(1):51–72, 1999.

[15] U. Fößmeier and M. Kaufmann. Drawing high degree graphs with low bend numbers. In F. J. Brandenburg, editor, *Proceedings of the 3rd International Symposium on Graph Drawing (GD '95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 254–266. Springer, 1996.

[16] A. Garg and R. Tamassia. A new minimum cost flow algorithm with applications to graph drawing. In S. C. North, editor, *Proceedings of the 4th International Symposium on Graph Drawing (GD '96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 201–213. Springer, 1996.

[17] C. Gutwenger, M. Jünger, G. W. Klau, S. Leipert, and P. Mutzel. Graph drawing algorithm engeneering with AGD. In diehl, editor, *Software Visualization, International Seminar, Dagstuhl Castle, Germany, May 20-25, 2001*, volume 2269 of *Lecture Notes in Computer Science*, pages 307–323. Springer, 2002. project home page at `http://www.ads.tuwien.ac.at/AGD`.

[18] C. Gutwenger, M. Jünger, S. Leipert, P. Mutzel, M. Percan, and R. Weiskircher. Advances in c-planarity testing of clustered graphs. In M. T. Goodrich and S. G. Kobourov, editors, *Proceedings of the 10th International Symposium on Graph Drawing (GD 2002)*, volume 2528 of *Lecture Notes in Computer Science*, pages 220–235. Springer, 2002.

[19] M. R. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *Journal of Computer and System Sciences*, 55:3–23, 1997. Special Issue on Selected Papers from STOC 1994.

[20] M. Jünger, S. Leipert, and M. Percan. Triangulating clustered graphs. Technical Report zaik2002-444, Zentrum für Angewandte Informatik Köln, 2002. `http://www.zaik.uni-koeln.de/\%7Epaper/preprints.html?show=zaik2002-444%`.

[21] G. W. Klau and P. Mutzel. Quasi orthogonal drawing of planar graphs. Technical Report MPI-I-98-1-013, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 1998. Available at `http://data.mpi-sb.mpg.de/internet/reports.nsf`.

[22] D. Lütke-Hüttmann. Knickminimales Zeichnen 4-planarer Clustergraphen. Master's thesis, Universität des Saarlandes, 1999. (Diplomarbeit).

[23] E. E. Moise. *Geometric Topology in Dimensions 2 and 3*, volume 47 of *Graduate Texts in Mathematics*. Springer, 1977.

[24] H. Nagamochi and T. Kameda. Constructing cactus representation for all minimum cuts in an undirected network. *Journal of the Operations Research Society of Japan*, 39(2):135–158, 1996.

[25] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16:421–444, 1987.

[26] R. Tamassia, G. Di Battista, and C. Batini. Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):61–79, 1988.