Karlsruhe Institute of Technology

Unpacking Planar Clustered Graphs: To Bend or not to Bend?

Bachelor Thesis of

Nina Zimbel

At the Department of Informatics
Institute of Theoretical Computer Science

Reviewers:	Prof. Dr. Dorothea Wagner
	Prof. Dr. Peter Sanders
Advisors:	Dr. Tamara Mchedlidze
	Marcel Radermacher, M. Sc.
	Dr. Ignaz Rutter

Time Period: 1st December 2016 – 31st March 2017

Statement of Authorship

I hereby declare that this document has been composed by myself and describes my own work, unless otherwise acknowledged in the text.

Karlsruhe, 31st March 2017

Abstract

Every time the layout of very big graphs shall be edited, it is wise to edit an abstraction instead and then transfer the changes back to the layout of the original graph. For abstraction we use a cluster-graph, in which each vertex represents one cluster of the original graph.

From a planar straight-line drawing of this cluster-graph we aim to construct a planar straight-line drawing of the original planar graph, which keeps the positions of the clusters and the embedding. To keep the positions of the clusters we limit the regions into which the clusters can be drawn by circles around the vertices of the cluster-graph. In contrast to some other papers in this field, which require biconnected clusters, we require connected clusters to be able to use our construction more flexibly.

Under the condition that the graph induced by two clusters does not contain another cluster in its interior, we show constructively that it is possible to generate a drawing of the original graph with the above-mentioned properties.

Deutsche Zusammenfassung

Immer dann, wenn das Layout sehr großer Graphen bearbeitet werden soll, ist es sinnvoll eine Abstraktion zu verwenden und dann die Änderungen auf das Layout des Originalgraphen zurück zu übertragen. Zur Abstraktion verwenden wir einen Clustergraphen, in dem jeder Knoten ein Cluster des Originalgraphen darstellt.

Das Ziel dieser Arbeit ist es, aus einer planaren, geradlinigen Zeichnung dieses Clustergraphen, unter Erhalt der Einbettung und der Positionen der Cluster, eine planare, geradlinige Zeichnung des planaren Originalgraphen zu erstellen. Zum Erhalt der Clusterpositionen werden die Flächen, in denen die Cluster gezeichnet werden dürfen, durch Kreise um die Knoten des Clustergraphen begrenzt. Im Gegensatz zu einigen anderen Arbeiten in diesem Bereich, die zweifach zusammenhängende Cluster vorraussetzen, verlangen wir einfach zusammenhängende Cluster, um diese Konstruktion flexibler einsetzen zu können.

Unter der Voraussetzung, dass der induzierte Graph zweier Cluster kein weiteres Cluster in seinem Inneren enthält, zeigen wir konstruktiv, dass es möglich ist eine Zeichnung des Originalgraphen mit den oben genannten Eigenschaften zu erstellen.

Contents

1. Introduction

Very big graphs can be found in many fields. Good examples are networks in social media like Facebook or Twitter, big wiring diagrams or road maps. The size of those graphs often prohibits any way of showing them on a display. So an abstraction is necessary. In most cases, these graphs are provided with a layer of meta data that specifies some kind of grouping on the vertices. These groups are called clusters and a graph with clusters is called *clustered graph*.
A *cluster-graph* of a clustered graph is an abstraction where each cluster is represented by one vertex and two vertices are adjacent if the two corresponding clusters are connected by edges.

Sometimes it is necessary to edit the layout of big graphs, for example if their layout is generated automatically and one wants to apply changes by hand. In this case, one can instead edit the layout of the abstraction, the cluster-graph. The problem is then to transfer the changes that were made in the layout of the cluster-graph to the layout of the original graph. This should be done in a way that does not destroy the editor's intentions and his perception of the whole graph. The cluster-graph should, in some way, be seen as a sketch for the drawing of the entire graph.

As this problem is very general, we restrict it to planar clustered graphs with planar cluster-graphs and straight-line drawings. Also we require the clusters and the whole clustered graph to be connected. We formalise the notion of preserving the user's sketch as follows: each cluster is required to lie in a small circle around the point representing this cluster in the cluster-graph (for an example see Figure 1.1).

1.1 Related Work

Planarity is a popular aspect in graph theory and network visualisation [11]. Fáry [8] shows that it is always possible to find a planar straight-line drawing of a planar graph. So when thinking about planar straight-line drawings, it is more interesting to consider variations of planarity. Many of those variations are studied [10], for example planarity of clustered graphs by defining c-planarity [3]. A graph is c-planar if it is planar, no two curves bounding clusters cross each other and no edge crosses the curve bounding a cluster more than once. In our setting, these curves are the circles into which the clusters shall be drawn.

Testing whether a graph is c-planar is a popular problem. Feng et al. [7] present an algorithm that determines whether a connected clustered graph is c-planar and, if it is,

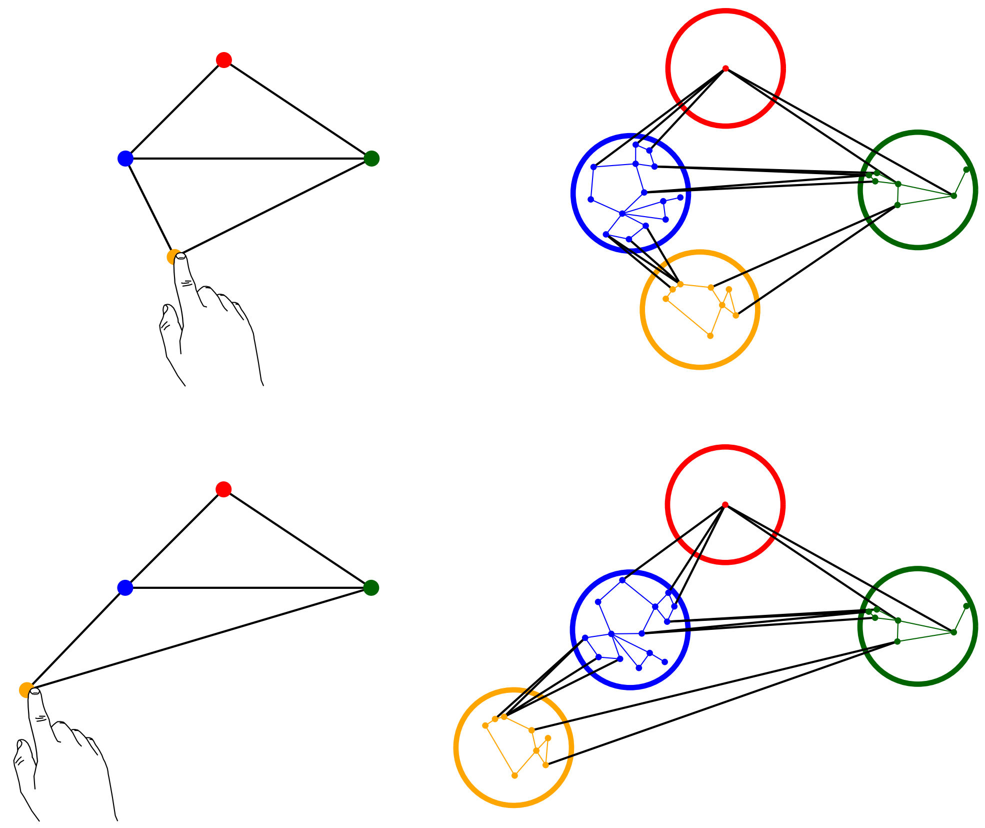

Figure 1.1: A small example of editing the layout of a planar graph with a clustering. In the left column the layout of the cluster-graph and in the right column the corresponding layout of the original graph is shown.

provides a c-planar embedding in $\mathcal{O}(n^2)$ time. It is an open problem if the existence of a c-planar drawing for a general graph can be tested in polynomial time and if a c-planar drawing for a general c-planar graph can be found in polynomial time [10].

Another popular question is, how much space the drawing of a graph needs. In [6] Feng et al. show that there exist clustered graphs whose straight-line drawings always require exponential area when the regions for the clusters are bordered by convex regions. They also present an algorithm which calculates a c-planar straight-line drawing of a clustered graph C, where the regions representing the clusters are convex, in $\mathcal{O}(n^{2.5})$ time. It considers nested clusters. Additionally they assume that the skeleton of C, which is the graph consisting of the boundary of each cluster and the edges between different clusters, is triconnected and that each of its clusters is biconnected.

Angelini et al. [2] introduce an algorithm that computes a c-planar straight-line drawing for every c-planar clustered graph for an arbitrary assignment of convex shapes to the clusters. They do not require a triconnected skeleton or biconnected clusters, but, in contrast to our problem, the algorithm takes no constraints for the drawing that restrict the positions of the clusters.

The problem of drawing hierarchical clustered graphs is investigated in [5] by Eades and Feng. They propose a method to visualise clustered graphs in 3-dimensional space. Each level of abstraction is drawn in a plane at a different z-coordinate. They show how to derive such a 3-dimensional drawing from a 2-dimensional c-planar drawing.

Alam et al. [1] examine rectangular maps in context of clustered graphs. A rectangular map is a rectangle that is subdivided into rectangular regions. Each rectangular region is assigned to one cluster and defines the region where this cluster can be drawn. The inter-cluster edges between two clusters must only pass through the borders of the rectangular regions that are shared by the rectangular regions of both clusters. It is required that the clustered graph is biconnected.
For this problem two condition are found: The first requires that each vertex has only inter-cluster neighbours of the same cluster and the second requires that each connected component of the cluster-graph has at most one cycle. Each of these conditions guarantees that a planar straight-line drawing on the given rectangular map is possible.

1.2 Our Contributions

We assume that a connected clustered graph C of a planar graph G and a planar straight-line drawing of the cluster-graph G_C of G is given as input. The requirement that the drawing of the cluster-graph is planar and straight-line is essential, as we get the positions of the clusters from this drawing. If it would not be planar it might not be possible to construct a straight-line drawing. If it would not be straight-line, we could get a situation where we have collinear clusters that we want to connect with straight edges. They would overlap and make it impossible for us to find a planar drawing of the original graph.
With this input we want to show that C has a c-planar straight-line drawing such that each cluster lies in a circle around the point that represents this cluster in the drawing of G_C.

We found that it is not always possible to achieve such a drawing. Consider a clustered graph where inside the interior of the subgraph induced by the vertices of two clusters a third cluster is embedded (see Figure 1.2). For the drawing we want to construct, we blow up the vertices of the cluster-graph to circles with uniform size. Inside of these, the clusters shall be drawn. But to keep the embedding of the original graph the red edge in Figure 1.2 has to be drawn “around” the enclosed cluster. This is not possible with a straight-line edge.

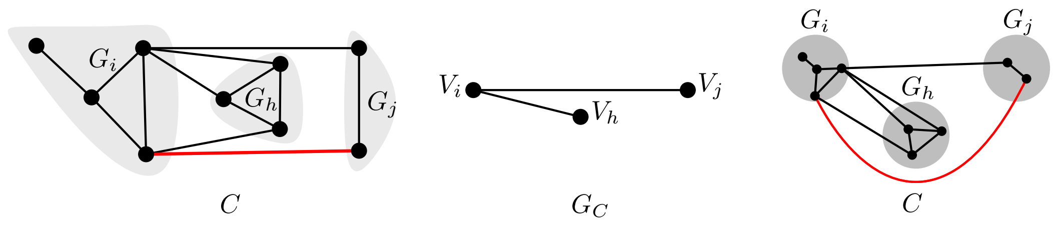

Figure 1.2: An example of a non-simple clustered graph. On the left the planar clustered graph C, in the middle the corresponding cluster-graph G_C with a planar straight-line drawing and on the right the constructed drawing of C with the desired properties, except that the red edge cannot be drawn straight-line.

To avoid such a situation, we define the property *simple* (see the preliminaries in Chapter 2) and require our input graph C to fulfil it.

Our approach is different to [5] and [6] as we restrict the clustered graph to have only one level of abstraction. Secondly, we do not restrict the cluster-graph to be tri- or biconnected, like in [1] or [6]. This is also the biggest challenge for our construction. And thirdly, we fix positions for the clusters, which distinguishes our problem from those of [2] and [6].

In the next chapter we introduce some definitions and notation. Chapter 3 includes our main theorem and its corollary with a constructive proof. With Chapter 4 we conclude this thesis.

2. Preliminaries

In this chapter we introduce definitions used in the next chapter. We mainly follow the definitions and notation of the paper "Fitting Planar Graphs on Planar Maps" by Alam et al. [1] for the basic concepts. During the whole chapter we use an example graph to illustrate the definitions.

Let $G = (V, E, \mathcal{E})$ be a planar embedded graph with a vertex set V, an edge set E and a plane embedding $\mathcal{E}$. Let V be partitioned into disjoint sets $\mathcal{V} = \{V_1, \ldots, V_k\}, k \geq 1$. We call the tuple $C = (G, \mathcal{V})$ a *planar clustered graph*. For each $i, 1 \leq i \leq k$, let $E_i \subset E$ be the set of edges between any two vertices of V_i. We call them *intra-cluster edges*. And let $E_{\text{inter}} \subseteq E$ be the set of all edges with endpoints in different clusters, the *inter-cluster edges*. See Figure 2.1 for an example of a planar clustered graph with inter-cluster edges highlighted in red. Note that $E = E_1 \uplus \cdots \uplus E_k \uplus E_{\text{inter}}$. We call $G_i = (V_i, E_i, \mathcal{E}_i), 1 \leq i \leq k$, *clusters* of G. The embedding $\mathcal{E}_i$ is given by restricting $\mathcal{E}$ to V_i.

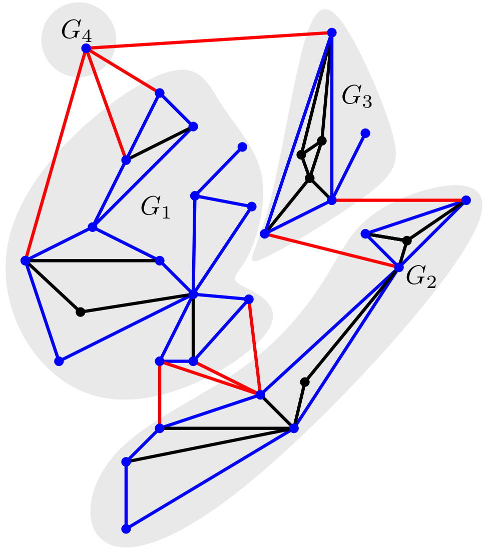

Figure 2.1: The clusters of the planar clustered graph are marked in gray and the inter-cluster edges in red. The boundaries of the outer faces of the clusters are marked in blue. The skeleton is the union of the red inter-cluster edges and the blue boundaries. The black vertices and edges form the interior of the clusters.

A clustered graph $C = (G, \mathcal{V})$ is *connected* (resp. *biconnected*) if G and each $G_i, 1 \leq i \leq k$, are connected (resp. biconnected) graphs.

Definition (Skeleton)**.** The *skeleton* $\mathcal{S}_C(G_i), 1 \leq i \leq k$, of a planar connected cluster G_i of G is defined as the boundary of the outer face of G_i. It consists of all vertices and edges that are incident to the outer face of G_i. The *skeleton* of a connected planar clustered graph C is $\mathcal{S}(C) = \bigcup_{i=1}^{k} \mathcal{S}_C(G_i) \cup E_{\text{inter}}$. See Figure 2.1 for illustration.

We observe that every $\mathcal{S}_C(G_i)$ is a *cactus graph*, a connected graph, in which every two cycles have at most one vertex in common and it is embedded in such a way that every vertex is adjacent to the outer face.

Definition (Block-cut Tree)**.** For every connected graph H, the *block-cut tree* $T(H)$ is constructed as follows (for illustration see Figure 2.2): We refer to the maximal biconnected components of H as *blocks*, to the vertices in one block B as $V(B)$ and to the edges of B as $E(B)$. The vertices of H that belong to more than one block are called *cut vertices*. The block-cut tree $T(H)$ has a *b-vertex* for each block and a *c-vertex* for each cut vertex. A b-vertex of $T(H)$ is connected to a c-vertex if and only if the block of the b-vertex contains the cut vertex corresponding to the c-vertex. Only vertices of different kinds are connected in $T(H)$.

We claim that the graph $T(H)$ is a tree. To see this, assume that $T(H)$ has a cycle. Then the subgraph of H belonging to this cycle would itself contain a cycle, which is biconnected. Therefore the division into blocks has not been maximal.
Let l be a leaf vertex of $T(H)$. We observe that l is always a b-vertex and refer to the block of l as *leaf block*.

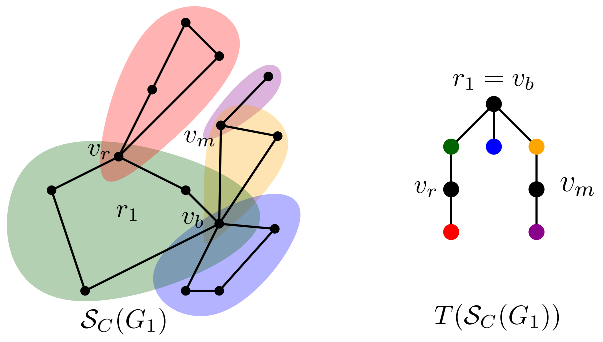

Figure 2.2: On the left side, the five blocks of $\mathcal{S}_C(G_1)$ of the example graph are marked. The cut vertex shared by the green, orange and blue block is chosen as root for the block-cut tree. On the right side the block-cut tree $T(\mathcal{S}_C(G_1))$ is depicted. The b-vertices are coloured in the same colors as the blocks on the left. C-vertices are shown in black. The green block is the parent block of the red block, and the yellow block is the parent block of the magenta block.

We create block-cut trees with an arbitrary c-vertex as root r_i for each $\mathcal{S}_C(G_i), 1 \leq i \leq k$. If $T(\mathcal{S}_C(G_i))$ has only one b-vertex, it has no c-vertices. In this case, we choose the vertex that shall be represented by the root c-vertex r_i out of the vertices of $\mathcal{S}_C(G_i)$ with at least one incident inter-cluster edge. Only if there is no such vertex, we choose a vertex with no incident inter-cluster edge. This is important for our construction for Theorem 3.1.
Let $p, q \in T(\mathcal{S}_C(G_i))$ be two b-vertices. If p and q are connected to the same c-vertex and p is closer to the root than q, we call the block represented by p the *parent block* of the block represented by q.

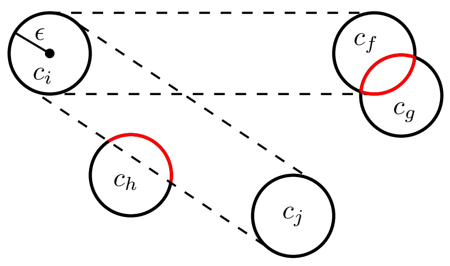

Figure 2.3: Circles shall neither overlap each other nor the strips (dashed lines) between the circles of adjacent clusters. This circle arrangement is not valid as c_h overlaps with the strip between c_i and c_j and c_f overlaps with c_g. Note that this circle arrangement can be made valid by reducing ϵ.

In the following we refer to blocks of a graph $\mathcal{S}_C(G_i)$ as *$\mathcal{S}_C$-blocks*. Note that every $\mathcal{S}_C$-block is a simple cycle or only consists of two vertices and one edge. In the latter case we call it *degenerated*. If $\mathcal{S}_C(G_i)$ only contains one vertex it has no $\mathcal{S}_C$-blocks. For a non-degenerated $\mathcal{S}_C$-block B, we call everything that is embedded inside the face, bounded by the cycle B in the graph C, the *interior* of B and denote the vertices of the interior of B with $V_\circ(B)$ and the edges with $E_\circ(B)$. Inside a degenerated $\mathcal{S}_C$-block B nothing can be embedded, so, in this case, the interior is empty and we define $V_\circ(B)$ and $E_\circ(B)$ as empty sets.

Definition (Inter-cluster Neighbours)**.** Let the *inter-cluster neighbours* $N_{\text{inter}}(v_i)$ of a vertex v_i in a cluster $G_j, 1 \leq j \leq k$, of a planar clustered graph C be the sequence of vertices $(x_1^i, \dots, x_l^i)$ in clusters other than G_j incident to v_i in clockwise order.
Let B be an $\mathcal{S}_C$-block in G_j. Let $(v_1, \dots, v_m), v_i \in V(B), i = 1 \dots m$, be the vertices of B in clockwise order. The inter-cluster neighbours of B are the inter-cluster neighbours of every v_i concatenated in clockwise order and without duplicates. We refer to them as $N_{\text{inter}}(B)$. To the inter-cluster edges incident to B we refer as $E_{\text{inter}}(B)$.

Definition (Circle Arrangement)**.** A *circle arrangement* $\mathbb{C}$ with respect to a clustered graph C with l clusters is a set of l circles in the plane with radius ϵ where one cluster is assigned to each circle. A circle arrangement $\mathbb{C}$ is *valid*, if the following two conditions hold:

- The circles do not overlap or touch each other.
- Consider *strips* of width 2ϵ that connect circles of adjacent clusters (see Figure 2.3). A strip connecting the circles c_i and $c_j, 1 \leq i, j \leq l$, is bordered by two parallel tangents. It consists of c_i, c_j and the region in between, enclosed by the tangents. We require that no circle c_h assigned to a cluster G_h overlaps or touches a strip between c_i and $c_j, i, j \neq h$.

Definition ($\mathbb{C}$-valid)**.** Let $\mathbb{C}$ be a valid circle arrangement. A drawing of a planar clustered graph C is $\mathbb{C}$-valid if every cluster is contained in its assigned circle.

For a $\mathbb{C}$-valid planar straight-line drawing of our example see Figure 2.4.

Definition (Simple)**.** A clustered graph $C = (G, \mathcal{V})$ is *simple* when, for every $1 \leq i, j \leq k$, there is no cluster $G_h, i, j \neq h$, embedded in the interior of the subgraph induced by $V_i \cup V_j$ (see Figure 2.5).

Note that this condition also implies that the clustering is *flat*, that means no cluster G_i contains another cluster in one of its inner faces. Figure 2.5 illustrates the case of a clustered graph with a flat clustering that is not simple. For a non-simple clustered graph C and a valid circle arrangement $\mathbb{C}$ with respect to C it is not possible to find a $\mathbb{C}$-valid planar straight-line drawing of C.

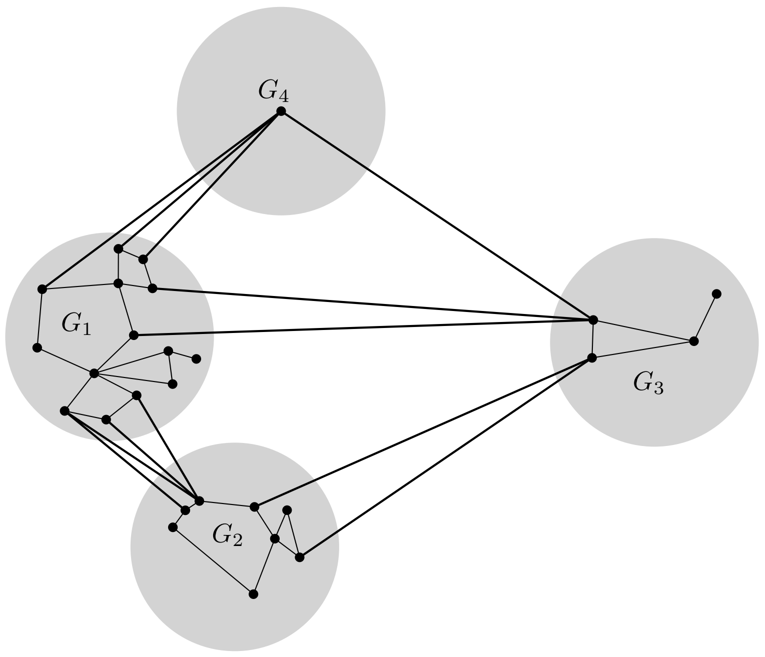

Figure 2.4: The gray circles depict a valid circle arrangement $\mathbb{C}$ with respect to the example graph. The drawing is $\mathbb{C}$-valid.

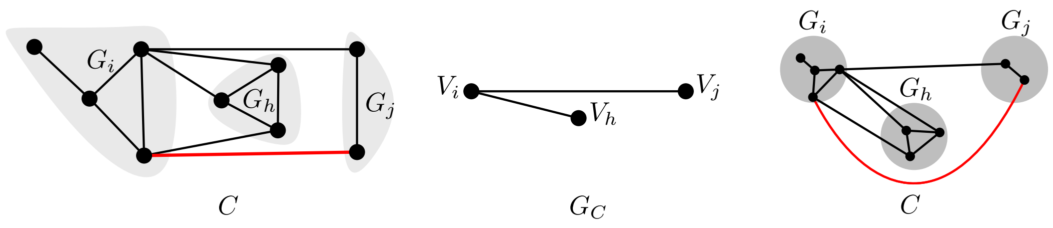

Figure 2.5: An example of a non-simple clustered graph C. On the left the clustered graph. The induced graph of $V_i \cup V_j$ contains a cluster G_h. In the middle the corresponding clustered-graph G_C with a planar straight-line drawing. On the right a valid circle arrangement $\mathbb{C}$ with respect to G_C in gray and a $\mathbb{C}$-valid planar straight-line drawing of C, except that the red edge can not be drawn straight-line without crossing G_h or changing the embedding.

Definition (Free)**.** Let B be a leaf $\mathcal{S}_C$-block of a cluster G_i of a connected planar clustered graph C. Consider two inter-cluster edges of $E_{\text{inter}}(B)$ that are incident to the same $u \in N_{\text{inter}}(B)$ of a cluster G_j. These two edges form, together with edges of B, a cycle in C. We call B *free* if none of these cycles contains any $\mathcal{S}_C$-block.

For an example of a block that is not free, see Figure 2.7.

Definition (Contracted Clustered Graph)**.** Let $G = (V, E, \mathcal{E})$ be a connected planar graph and let $C = (G, \mathcal{V})$ be a connected clustered graph. Let B be a free leaf $\mathcal{S}_C$-block of $T(\mathcal{S}_C(G_i))$. The *contracted clustered graph* $C/B = (G/B, \mathcal{V}')$ is constructed from C as follows (for an example see Figure 2.6):
The graph $G/B = (V', E', \mathcal{E}')$ is constructed by removing the interior of B, replacing the vertices of B by one single vertex v, eliminating double edges and removing loops. If B has a parent $\mathcal{S}_C$-block B_p, let v be the cut-vertex that B shares with B_p, else let v be the vertex represented by the root r_i of the block-cut tree $T(\mathcal{S}_C(G_i))$. The set of vertices of G/B is $V' = (V \setminus V(B) \setminus V_\circ(B)) \cup \{v\}$ and the set of edges $E' = (E \setminus E(B) \setminus E_\circ(B) \setminus N_{\text{inter}}(B)) \cup \{(v, u) | u \in N_{\text{inter}}(B)\}$. The embedding $\mathcal{E}'$ is obtained from $\mathcal{E}$ by ordering the edges (v, u) according to the cyclic order of $N_{\text{inter}}(B)$. In C/B the set of vertices of cluster G_i is now $V_i' = V_i \setminus V(B) \setminus V_\circ(B) \cup \{v\}$ and $\mathcal{V}' = \{V_1, \ldots, V_{i-1}, V_i', V_{i+1}, \ldots, V_k\}$.

Proof. We have to show that the embedding $\mathcal{E}'$ obtained by the construction is always well-defined. During the contraction, the edges of $N_{\text{inter}}(B)$ that are incident to the same vertex $u \in N_{\text{inter}}(B)$ are replaced by one single edge (v, u). Therefore we have to show that these edges are embedded consecutively around B and around u. Otherwise the resulting ordering of vertices around u and v would not be well-defined.
The requirement for B to be free guarantees that all edges of $E_{\text{inter}}(B)$ incident to the same vertex $u \in N_{\text{inter}}(B)$ are ordered consecutively around u. Additionally, the embedding of C is planar and B is a leaf block. This means that edges to the same inter-cluster neighbours of B are ordered consecutively around B. So we see that the embedding $\mathcal{E}'$ is well-defined. □

We refer to the operation of constructing a contracted clustered graph C/B as *contracting* B (into the vertex v). When constructing C from the contracted clustered graph C/B we *unpack* B.

To construct the *cluster-graph* $G_C = ((\mathcal{V}, E_C, \mathcal{E}_C), \mathcal{V})$ of C, consider all $\mathcal{S}_C$-blocks of a planar clustered graph $C = (G, \mathcal{V})$. Contract the free leaf $\mathcal{S}_C$-blocks iteratively. The resulting clustered graph, where all blocks are contracted and each cluster contains only one vertex, is G_C (see Figure 2.8). We claim that there is always at least one free $\mathcal{S}_C$-block in a connected planar clustered graph. Therefore it is always possible to contract all $\mathcal{S}_C$-blocks of a connected planar clustered graph C.

If an $\mathcal{S}_C$-block B of cluster G_i is not free, there is at least one $\mathcal{S}_C$-block A incident to an inter-cluster neighbour u of B that is the reason for that. This $\mathcal{S}_C$-block A lies inside the cycle formed by two inter-cluster edges between B and u and edges of B. Only if there is another $\mathcal{S}_C$-block A' inside this cycle in cluster G_i it is possible that A itself is non-free. As A' has to lie inside the cycle it cannot coincide with B. So the only situation where C has no free $\mathcal{S}_C$-blocks is when there is an infinite number of $\mathcal{S}_C$-blocks in C.

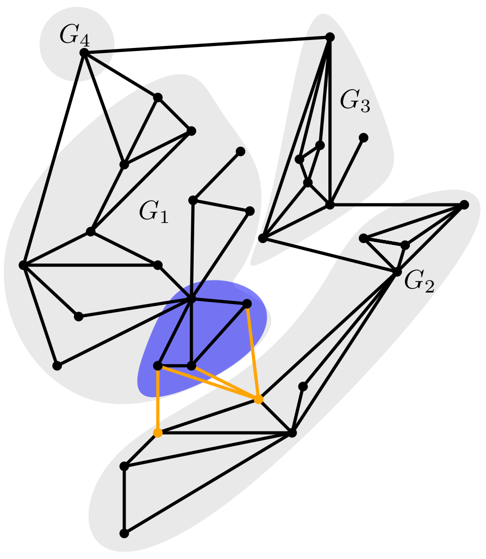

Figure 2.6: On the left the planar clustered graph C. One free leaf $\mathcal{S}_C$-block B (and its interior) of cluster G_1 is highlighted in blue. Its inter-cluster neighbours are highlighted in orange. On the right the contracted clustered graph C/B is depicted. The cut-vertex that B shares with its parent block in C is marked in blue.

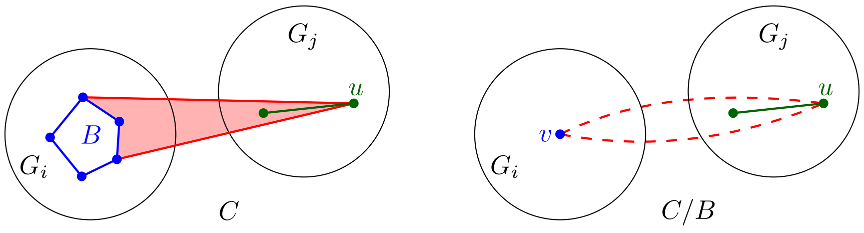

Figure 2.7: On the left a clustered graph C with a non-free $\mathcal{S}_C$-block B that shall be contracted. The cycle formed by the two red inter-cluster edges contains an $\mathcal{S}_C$-block of G_j. The embedding of the contracted graph C/B is not well-defined.

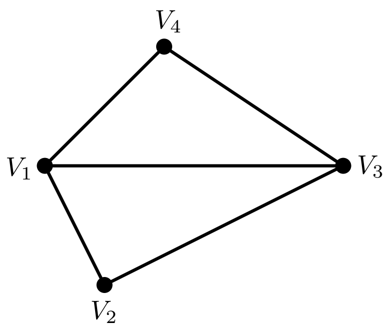

Figure 2.8: The cluster-graph G_C of our example graph.

Clustered graphs can be drawn in such a way that the clusters are bounded by closed curves. A drawing of a clustered graph is *c-planar* if it is planar, no two curves bounding clusters cross each other and no edge crosses the curve bounding a cluster more than once. In our setting, these curves are the circles of the circle arrangement. A graph is *c-planar* if there is a c-planar drawing of it. We observe that every $\mathbb{C}$-valid planar straight-line drawing is c-planar.

With these definitions we are now able to formulate the main theorem of this thesis in the next chapter.

3. Unpacking Planar Clustered Graphs

In this chapter we present the main theorem of this thesis and its corollary. Theorem 3.1 shows that we can create a drawing of a simple connected planar clustered graph C, with the desired properties, based on a drawing of the graph C/B, where the $\mathcal{S}_C$-block B is contracted. Our main result is Corollary 3.2, which shows that not only one $\mathcal{S}_C$-block can be unpacked, but, based on the cluster-graph, that all blocks of a clustered graph can be unpacked.

Theorem 3.1. *Let $G = (V, E, \mathcal{E})$ be a connected planar graph with a plane embedding $\mathcal{E}$. Let $C = (G, \mathcal{V})$ be a simple connected clustered graph. Let B be a free leaf $\mathcal{S}_C$-block of a cluster G_i of G. Let $C/B = (G/B, \mathcal{V}')$ be a contracted clustered graph with $G/B = (V', E', \mathcal{E}')$ and $\mathbb{C}$ a circle arrangement with respect to C/B. Let C/B have a $\mathbb{C}$-valid planar straight-line drawing with an embedding identical to $\mathcal{E}'$.*
Then C has a $\mathbb{C}$-valid planar straight-line drawing with an embedding identical to $\mathcal{E}$.

Our main Theorem 3.1 shows that we can "reverse" the contraction of one block while keeping a $\mathbb{C}$-valid planar straight-line drawing. As we can do this, we can also reverse all contractions that we had to do to receive G_C from C. So the following is a corollary to Theorem 3.1. It is implied by Theorem 3.1 by induction over the blocks in the clustered graph.

Corollary 3.2. *Let $G = (V, E, \mathcal{E})$ be a connected planar graph with a plane embedding $\mathcal{E}$. Let $C = (G, \mathcal{V})$ be a simple connected clustered graph and let $G_C = (\mathcal{V}, E_C, \mathcal{E}_C)$ be the cluster-graph of C. Let Γ_C be a planar straight-line drawing of G_C with an embedding identical to $\mathcal{E}_C$.*
Then there is a circle arrangement $\mathbb{C}$ so that Γ_C is $\mathbb{C}$-valid and C has a $\mathbb{C}$-valid planar straight-line drawing with an embedding identical to $\mathcal{E}$.

Proof. The statement of this corollary follows by induction on the $\mathcal{S}_C$-blocks of C from Theorem 3.1. Note that each contracted clustered graph with l blocks that originates from C by contracting an arbitrary number $b, 0 \leq b \leq l$, of blocks has the same blocks as C, except that the contracted ones are missing.
In the base case, all blocks are contracted. This is the graph G_C with the drawing Γ_C. It is easy to see that for every planar straight-line drawing of a cluster-graph, where every cluster contains only one vertex, a valid circle arrangement $\mathbb{C}$ exists.

With $\mathcal{H}_b$ we denote the set of all simple connected planar clustered graphs that originate from C by contracting b free $\mathcal{S}_C$-blocks. Their embeddings are fixed by the contractions. The induction hypothesis states, that every $H_b \in \mathcal{H}_b$ with its embedding $\mathcal{E}_b$ has a $\mathbb{C}$-valid planar straight-line drawing with an embedding identical to $\mathcal{E}_b$.
Now consider the graph $H_{b-1} \in \mathcal{H}_{b-1}$ with its embedding $\mathcal{E}_{b-1}$. When contracting a free leaf $\mathcal{S}_{H_{b-1}}$-block B, we get the graph $H_{b-1}/B \in \mathcal{H}_b$. Let $\mathcal{E}'$ be its embedding. By induction hypothesis, H_{b-1}/B has a $\mathbb{C}$-valid planar straight-line drawing with an embedding identical to $\mathcal{E}'$. By Theorem 3.1, this drawing can be extended to a $\mathbb{C}$-valid straight-line drawing of H_{b-1} with an embedding identical to $\mathcal{E}_{b-1}$. $\square$

Before we proceed to the proof of Theorem 3.1 in Section 3.1, we present several definitions and facts we need later on.

Let (v, w) be an edge of a graph G, let w have a fixed position in the plane and let c be a circle in the plane that does not contain w. We want to draw (v, w) in such a way that v lies in c. We define cone(w, c) as the region where (v, w) can be drawn with these conditions. It is bordered by two tangents to c passing through w and the circular arc of c that connects the two tangent points (see Figure 3.1).

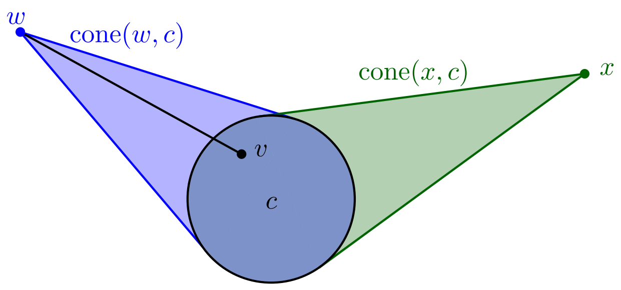

Figure 3.1: An edge (v, w), where w has a fixed position outside of c and v lies anywhere in c, is drawn inside cone(w, c).

Let $\mathbb{C}$ be a circle arrangement with respect to a clustered graph C. We refer to a circle of $\mathbb{C}$ assigned to a cluster G_i of C as *ϵ-circle of G_i*.

Observation 3.3. Let $v \in V_i, w \in V_j, i \neq j$, with $(v, w) \in E_{\text{inter}}$ and let c be a circle that lies inside the ϵ-circle of G_i. Then cone(w, c) lies completely inside the strip between G_i and G_j (see Figure 3.2).

Assume that we have a $\mathbb{C}$-valid planar straight-line drawing of the clustered graph C. Now we want to show that we can construct a circle c, with "nice" properties, around any vertex $v \in \mathcal{S}(C)$. In the proof of Theorem 3.1 we need such a circle around the vertex into which the $\mathcal{S}_C$-block B is contracted.

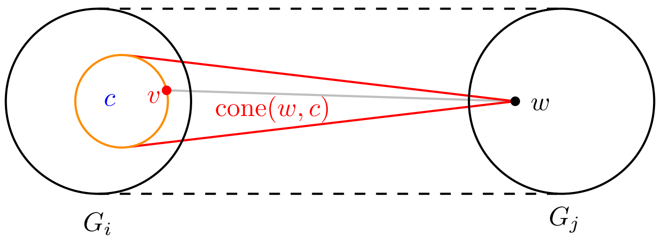

Figure 3.2: The dashed lines mark the strip between the ϵ-circles of the clusters G_i and G_j. For every position of c in the ϵ-circle of G_i and w in the ϵ-circle of G_j, the cone cone(w, c) lies completely inside the strip connecting G_i and G_j.

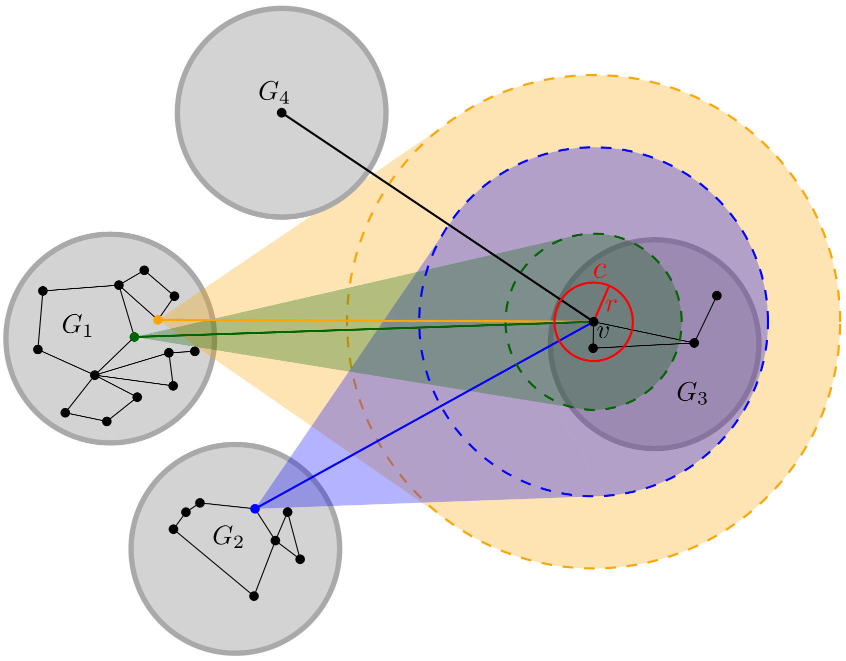

Figure 3.3: Cluster G_3 contains vertex v. The cones of the inter-cluster edges of v are marked in different colours. The ϵ-circles for each cluster are marked by circles with thick borders. The circle c with radius r, which fulfils both properties demanded by Lemma 3.4, is marked in red.

Lemma 3.4. *Let $G = (V, E, \mathcal{E})$ be a connected planar graph with a planar embedding $\mathcal{E}$. Let $C = (G, \mathcal{V})$ be a simple connected planar clustered graph and $\mathbb{C}$ a circle arrangement with respect to C. Let v be a vertex of $\mathcal{S}(C)$ in a cluster G_i and let Γ be a $\mathbb{C}$-valid planar straight-line drawing of $\mathcal{S}(C)$.*
Then there is a circle c with a radius $r > 0$ centered at the position of v in Γ with the following properties:

(I) It lies properly inside the ϵ-circle of G_i.

(II) Shifting v to any position inside c does not cause any edge crossings in the clusters of $\mathcal{S}(C)$ except of G_i, and keeps the ordering of the inter-cluster edges around v.

Proof. The two properties demanded by the lemma can be translated directly into the following two requirements for r. For illustration see Figure 3.3.

(I) **ϵ-circle**: The radius r has to be small enough so that c lies properly inside the ϵ-circle of cluster G_i. Let o_ϵ be the center of the ϵ-circle of G_i and o_c be the center of c, which is the position of v. Then the condition for r is: $r < \epsilon - \|o_\epsilon - o_c\|$.

(II) **Cones**: Let G_h be a cluster of $\mathcal{S}(C)$ that is connected to v by an inter-cluster edge (u, v). Note that such a cluster does not need to exist. If v has no incident inter-cluster edges, there is no condition of this kind. If $\text{cone}(u, c)$ does not overlap the drawing of $\mathcal{S}_C(G_h)$, then, for any position of v in c, the edge (u, v) does not cross the drawing of G_h. So we set r_u to be the radius of a circle c_u around v that is small enough that $\text{cone}(u, c_u)$ does not intersect the drawing of G_h. For illustration see Figure 3.4. We choose $r \le \min(\{r_u \mid u \in N_{\text{inter}}(v)\})$.

From both constraints we get possible intervals for r. We choose r such that it is in the intersection of both intervals. For an example construction see Figure 3.3. Note that it might be useful to choose a big r from this interval to get "nicer" drawings.

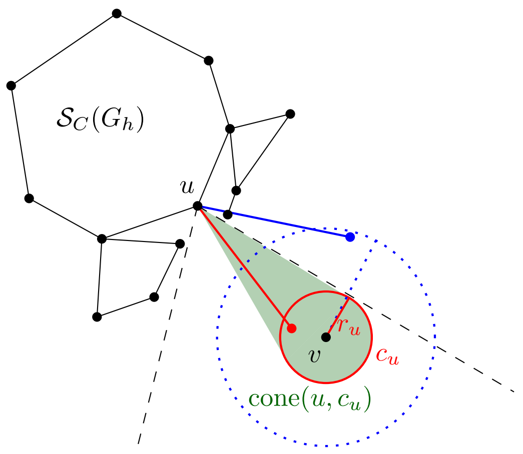

Figure 3.4: The maximal circle centered at v, where $\text{cone}(u, c_u)$ does not cross components of $\mathcal{S}_C(G_h)$ is marked in red. The dashed lines mark the maximum region for $\text{cone}(u, c_u)$ (green). Shifting v inside c_u causes edges like the red one inside $\text{cone}(u, c_u)$; shifting v inside the blue circle can cause edges like the blue one, which crosses components of $\mathcal{S}_C(G_h)$.

In the following we show that the resulting radius r for the circle c is always greater than zero. Constraint (I) cannot shrink r to zero, because the position of v is properly inside the ϵ-circle of G_i (not on its border). What is left to show, is that all radii r_u of constraint (II) are greater than zero.

Let (u, v) be an inter-cluster edge. Let $x, y \in V \setminus \{u, v\}$ and $(x, y) \in E$. The edge (x, y) does not intersect (u, v) in Γ, as Γ is a planar drawing. So there is a small space around (u, v) where $\text{cone}(u, c)$ lies. This guarantees that the borders of $\text{cone}(u, c)$ do not cross v and that r_u is greater than zero. As this is true for all cones and v is positioned in the center of all c_u, the minimum of $\{r_u \mid u \in N_{\text{inter}}(v)\}$ is also greater than zero.

Now we argue that both properties demanded by the lemma are fulfilled by the constructed circle c. Property (I) is fulfilled because of constraint (I), which guarantees that r is not bigger than the minimal distance of v to the border of the ϵ-circle. Property (II) is fulfilled because of constraint (II). It guarantees that the cones $\{\text{cone}(u, c) \mid u \in N_{\text{inter}}(v)\}$ can only contain G_i and the vertices in $N_{\text{inter}}(v)$. For any position of v in c, all inter-cluster edges incident to v have to lie inside these cones when they are drawn straight-line. So the first part of property (II) is fulfilled. What is left to show, is that the circular ordering of the inter-cluster edges around v remains the same for every position of v inside c.

Let $a, b \in N_{\text{inter}}(v), a \neq b$ and let G_a be the cluster of a. If the segment between a and b crosses c, the relative position of a to b cannot be changed by moving v inside c (see Figure 3.5). So assume that the segment does not cross c.
To change the circular order of $N_{\text{inter}}(v)$ around v, the line passing through the positions of a and b has to cross c. W.l.o.g. let b be nearer to v than a. Then b has to lie inside $\text{cone}(a, c)$. By construction, $\text{cone}(a, c)$ does not contain any vertices of the cluster G_a other than a. Also $\text{cone}(a, c)$ lies completely inside the strip between G_i and G_a (see Observation 3.3). So no vertices other than a and vertices of G_i lie inside $\text{cone}(a, c)$. This means b cannot lie inside $\text{cone}(a, c)$ and the line through a and b cannot pass through c. Therefore, the circular ordering of $N_{\text{inter}}(v)$ around v is preserved.

□

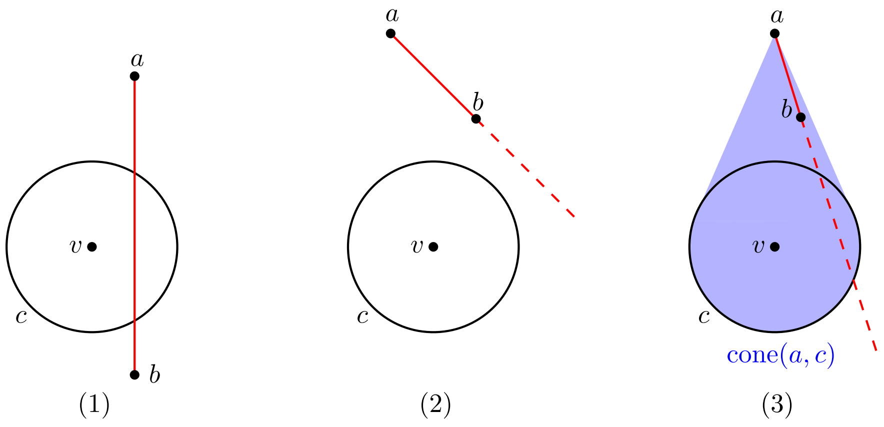

Figure 3.5: Different relative positions that two inter-cluster neighbours a and b of v can have regarding c: (1) The segment $\overline{ab}$ crosses c; the circular order does not change inside c. (2) The line through a and b does not cross c; the circular order does not change inside c. (3) The vertex nearer to c (here b) lies inside the cone of the other (here a); the circular order changes inside c.

3.1 Construction of the Drawing

In this section we give a construction of the resulting drawing of Theorem 3.1 and in the next section we prove that it fulfils all properties demanded by the theorem. The theorem states:

Theorem 3.1. *Let $G = (V, E, \mathcal{E})$ be a connected planar graph with a plane embedding $\mathcal{E}$. Let $C = (G, \mathcal{V})$ be a simple connected clustered graph. Let B be a free leaf $\mathcal{S}_C$-block of a cluster G_i of G. Let $C/B = (G/B, \mathcal{V}')$ be a contracted clustered graph with $G/B = (V', E', \mathcal{E}')$ and $\mathbb{C}$ a circle arrangement with respect to C/B. Let C/B have a $\mathbb{C}$-valid planar straight-line drawing with an embedding identical to $\mathcal{E}'$.*
Then C has a $\mathbb{C}$-valid planar straight-line drawing with an embedding identical to $\mathcal{E}$.

Let Γ' be the $\mathbb{C}$-valid planar straight-line drawing of C/B. In the following we construct the $\mathbb{C}$-valid planar straight-line drawing Γ of C. Let v_c be the vertex into which B is contracted in C/B. To obtain the drawing Γ of the unpacked graph, we determine the positions for the newly added set of vertices $(V(B) \cup V_\circ(B)) \setminus \{v_c\}$. All vertices of the contracted clustered graph $C/B = (G/B, \mathcal{V}')$ will stay in the positions they had in Γ'. Let c be a circle around v_c with the following two properties, which exist by Lemma 3.4:

(I) It lies properly inside the ϵ-circle of G_i.

(II) Shifting v to any position inside c does not cause any edge crossings in the clusters of $\mathcal{S}(C)$ except of G_i, and keeps the ordering of the inter-cluster edges around v.

Now the task is to find a convex shape inside c where we can safely place the new vertices on the border. We will choose an eye-like shape σ that does not overlap with those parts of the circle that are "occupied" by components of G_i. After placing the vertices on σ the cycle B describes a convex polygon. So we can then apply the algorithm for drawing a graph with prescribed convex outer face by Chambers et al. [4].

In the contracted clustered graph C/B, the set of inter-cluster edges E_B that are incident to B in C are incident to v_c in C/B and are embedded consecutively around v_c. We define the *bordering edges* of a free leaf $\mathcal{S}_C$-block as follows:

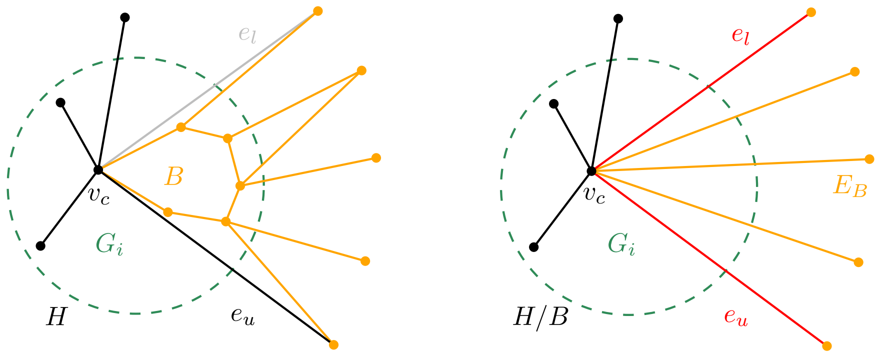

Figure 3.6: On the left v_c and B of the graph H and on the right H/B with the cut vertex v_c where B is contracted into. The bordering edges e_l and e_u are coloured in red in the drawing Γ' of H/B and the remaining inter-cluster edges are coloured in orange. In the drawing of H the position e_l has in Γ' is shown in gray. In this example e_u does also exist in H. Because of this it is shown in black. The ϵ-circle of the cluster G_i of B is she dashed green circle.

Definition (Bordering Edges)**.** Let B be a free leaf $\mathcal{S}_H$-block of a clustered graph H. Let v_c be the vertex into which B is contracted in H/B. Let E_B be the set of inter-cluster edges, which originate from replacing the inter-cluster edges incident to vertices of $V(B) \setminus \{v_c\}$ in H by inter-cluster edges incident to v_c in H/B, when contracting B.
Then we call the first and the last edge of E_B in clockwise order around v_c in H/B *lower* and *upper bordering edge* to B. We denote them by e_l and e_u.

Note that there are no bordering edges of an $\mathcal{S}_H$-block, if $|E_B| = 0$ or if v_c has no other incident edges than those in E_B. Also it is possible that $e_l = e_u$, if $|E_B| = 1$.

Let e_l and e_u be the lower and upper bordering edge of B respectively, if they exist (see Figure 3.6). They may not exist in C. So, to be able to use them in the construction of Γ, we store their positions. We consider the pie slice of c bordered by e_l and e_u in Γ' that contains the edges E_B. Into this pie slice we place σ.

If $|E_B| = 0$, there are no bordering edges of B and no inter-cluster edges incident to it. When B is the only $\mathcal{S}_C$-block of $\mathcal{S}_C(G_i)$ nothing else is inside c, so we do not need to construct σ. We just embed the cycle B on c. Then we apply the algorithm for drawing a graph with a prescribed convex outer face of Chambers et al. [4] to complete the drawing of B with its interior and we are done. Note that this can only happen if C only consists of B and its interior. So in the following we only consider vertices v_c with edges that belong to components of C other than B or its interior.
If $|E_B| = 0$ and B has a parent block B_p, we consider the edges incident to v_c that precede and follow the inter-cluster edges of B incident to v_c. The last edge preceding B and the first edge following B in clockwise order also include a pie slice of c opposite to B_p. We call these edges f and g. Note that f and g can coincide if B is degenerated.

Also there are no bordering edges of B if v_c has no other incident edges than those of E_B. This can only happen when B is the only block in G_i. Otherwise B has a parent block B_p and v_c has incident intra-cluster edges belonging to B_p. So, in this case, v_c is the vertex represented by the root r_i of the block-cut tree $T(\mathcal{S}_C(G_i))$. We chose v_c from the vertices of B in such a way that it has at least one incident inter-cluster edge if there is such a vertex in B. If v_c has an incident inter-cluster edge, this edge is not in E_B and it is a bordering edge to B. If there is no vertex in B that has incident inter-cluster edges, then we are in the case that $|E_B| = 0$ and B is the only $\mathcal{S}_C$-block of $\mathcal{S}_C(G_i)$ (see above). Therefore we do not have to consider the case that v_c has only incident edges of E_B in the following.

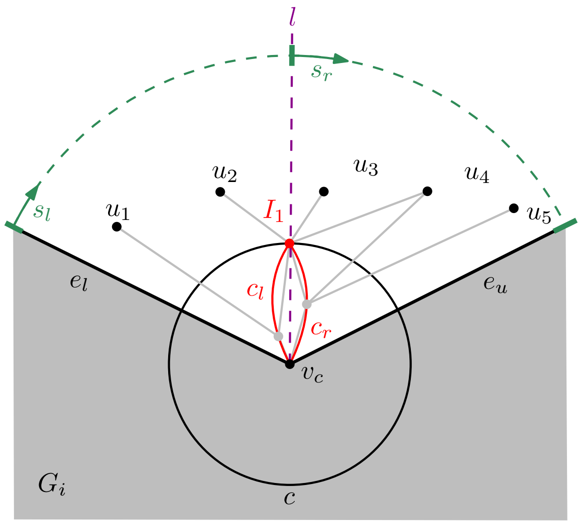

Figure 3.7: Illustration for the case where B has bordering edges. The fat black edges e_l and e_u are the bordering edges. In gray there is an example constellation for the vertices of B with inter-cluster edges. The sequences are $s_l = (u_1, u_2)$ and $s_r = (u_3, \dots, u_5)$ in this example. The eye-like shape σ is marked in red and the region where components of G_i other than B or its interior can lie is coloured in dark gray.

The cases we need to consider for the construction of σ are, firstly, that there are bordering edges of B and, secondly, that $|E_B| = 0$ but B has a parent block B_p. Next we show how to determine the two extreme points of σ and then we construct the two convex curves bounding σ.

Let l be the bisector through the pie slice bordered by e_l and e_u, or f and g respectively (see Figures 3.7 and 3.8). If l crosses any inter-cluster neighbours of B, rotate it a small angle to avoid this. The two half-planes h_l and h_r induced by l partition the inter-cluster neighbours of B into the two sequences $s_l = (u_1, \dots, u_q)$ and $s_r = (u_{q+1}, \dots, u_m)$. The vertex v_c and the intersection I_1 of l with c are the two extreme points of σ. We construct an eye-like shape by connecting the two extreme points with two curves, c_l and c_r, that do not intersect or touch the bordering edges of B, except in v_c.

If we constructed l between f and g, it can cross through components of G_i before crossing c (see Figure 3.8). Because of this, we construct a new circle c' inside c that does not contain any components of G_i, except of v_c and parts of f and g. This is always possible as between each two vertices of a drawing there is always some free space. Then I_1 is the intersection of l with c'.

Before we continue with the construction of σ, we make an observation that helps us to understand the given graph structure better.

Claim. *Let B be a non-degenerated free leaf $\mathcal{S}_H$-block of a simple connected clustered graph H with embedding $\mathcal{E}$. Let v_c be the vertex into which B is contracted in the contracted clustered graph H/B with embedding $\mathcal{E}'$, which is implied by the contraction of B. Let v_c have edges that belong to components of C other than B or its interior. Let $(v_1, \dots, v_m)$ be the vertices $V(B) \setminus \{v_c\}$ in clockwise order starting with the first vertex after v_c. Let $s_l = (u_1, \dots, u_q)$ and $s_r = (u_{q+1}, \dots, u_m)$ be two sequences of inter-cluster neighbours in clockwise order around v_c of B that can be concatenated to the whole sequence $N_{\text{inter}}(B)$ in the following way: $s_l \cdot s_r = N_{\text{inter}}(B)$. Let $v_i, v_j \in V(B) \setminus \{v_c\}, i < j$ and let H/B have a planar straight-line drawing.*

In H, if v_i is connected to a vertex of s_r, then v_j can only be connected to vertices of s_r. If, the other way around, v_j is connected to a vertex of s_l, then v_i can only be connected to vertices of s_l.

Proof. In the graph H, let v_i be connected to a vertex u_r of s_r and v_j be connected to a vertex u_l of s_l. In H/B, the cut vertex v_c is connected to u_l and u_r. Let Γ' be the planar and straight-line drawing of H/B. The contraction of B and the embedding of H dictate that (v_c, u_r) comes before (v_c, u_l) in clockwise order around v_c starting from the lower bordering edge of B. So the region where (v_c, u_l) can be embedded in Γ' is restricted by (v_c, u_r) and other edges incident to v_c. But u_l is not in this region. So Γ' can either not be a straight-line drawing or it is not planar. (For illustration see Figure 3.9.) $\Diamond$

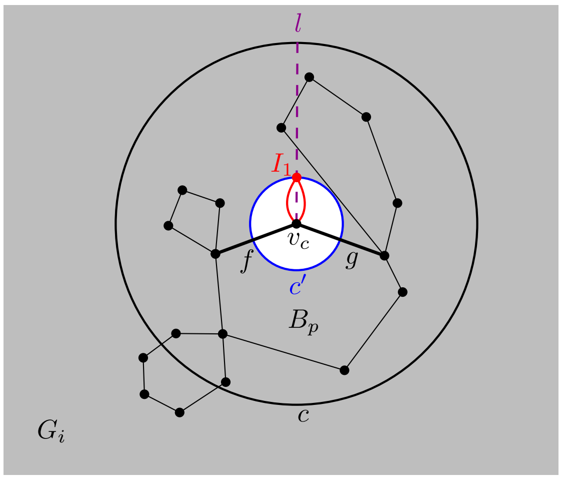

Figure 3.8: Illustration for the case where B has no bordering edges and therefore no incident inter-cluster edges. A circle c' is constructed inside c. The eye-like shape σ is marked in red and the region where components of G_i other than B or its interior can lie is coloured in dark gray.

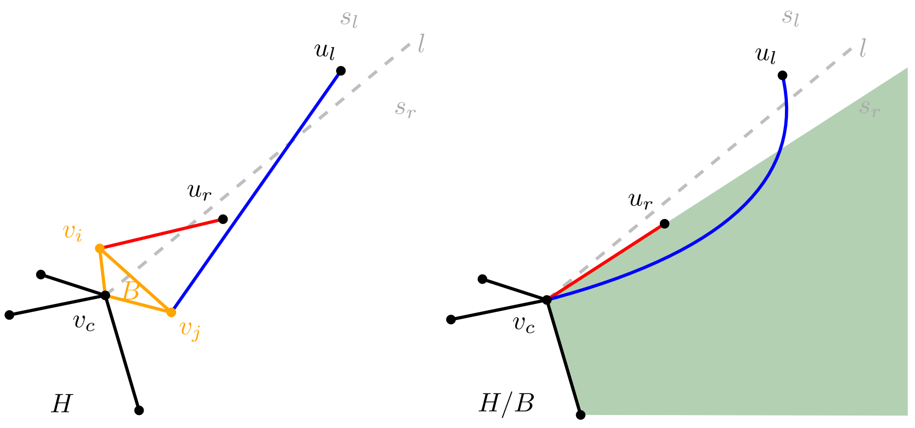

Figure 3.9: The dashed line l divides the inter-cluster neighbours of B into two sequences s_l and s_r. On the left the situation in H is illustrated. On the right the drawing of H/B is shown. The region where (v_c, u_l) can be embedded straight-line in the drawing of H/B with the given embedding $\mathcal{E}'$ is marked in green. The vertex u_l is not inside this region. So a planar straight-line drawing with embedding $\mathcal{E}'$ is not possible.

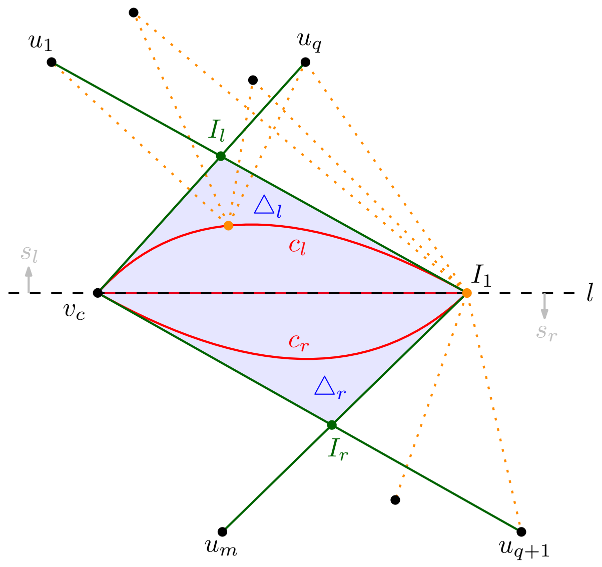

Figure 3.10: The black dots on the top mark the vertices of s_l, those on the bottom the vertices of s_r. In blue one can see Bézier curves c_l and c_r. In orange one can see two example vertices. One on c_l, which is connected to all vertices of s_l, and one on I_1, which is connected to all vertices of s_l and s_r. Note that they cannot exist both.

From the claim we see that there can be at most one vertex that is connected to vertices from s_l and s_r. We place this vertex on I_1, if it exists. All other vertices that are connected either only to vertices of s_l or only to vertices of s_r are placed on the corresponding curve c_l or c_r.

Each $v \in V(B)$ has a subset $U_v \subseteq N_{\text{inter}}(B)$ of adjacent inter-cluster neighbours. Let $a, b \in V(B) \setminus \{v_c\}, a \neq b$ such that a is between v_c and b on the cycle B in clockwise order. The partitioning into the two sequences s_l and s_r in the claim is arbitrary. So we can consider another partitioning into s_a and s_b. Let s_a be the sequence $(u_1, \ldots, u_a)$, where u_a is the last inter-cluster neighbour of a in clockwise order. Then $U_{v_a} \cap U_{v_b}$ can at most contain u_a. We see that all sets U_v include consecutive sequences of vertices and that two of them can at most have one vertex in common.

In the following, we explain how exactly the curves c_l and c_r have to be constructed in order to avoid crossings among edges incident to B and edges of B itself in Γ.
To determine c_l and c_r, we construct triangles $\triangle_l$ and $\triangle_r$ left and right of l. The triangles $\triangle_l$ and $\triangle_r$ share one side, which lies on l connecting v_c and I_1 (see Figure 3.10). The third corner I_l of $\triangle_l$ is the intersection of $\overline{I_1 u_1}$ and $\overline{v_c u_q}$. Likewise, the third corner I_r of $\triangle_r$ is the intersection of $\overline{I_1 u_m}$ and $\overline{v_c u_{q+1}}$. This construction is only valid if the segments, which determine the third corners of the triangles do not cross l. For the proof that this cannot happen, see proof (4a) in the next section.
If s_l is empty, we do not need to place vertices with incident inter-cluster edges on c_l, but there might still be other vertices to place there. In this case, we define I_l to be a point a little bit left of I_1 on c (or c' respectively). This guarantees that c_l does not cross any components of G_i except of v_c. We do an analogous construction for c_r, if s_r is empty.

The curves c_l and c_r have to lie inside c (or c' respectively) and their corresponding triangle. For the construction with c' this is always given as B has no inter-cluster neighbours in this case. If $\triangle_l \cap c \neq \triangle_l$, replace the triangle $\triangle_l$ by a triangle inside $\triangle_l$ that fulfils this condition. Such a triangle does always exist as $\triangle_l \cap c \neq \emptyset$ and both, $\triangle_l$ and c, contain v_c and I_1. We handle $\triangle_r$ equivalently. Additionally, c_l and c_r have to be strictly convex, be monotone with respect to $\overline{v_c I_1}$ and connect I_1 and v_c. If all this is given, also σ is strictly convex.

Types of curves that satisfy all the mentioned constraints are for example parabolas that stay inside the corresponding triangle and pass through v_c and I_1 or Bézier curves of second degree with v_c, I_1 and one third, suitable control point inside the corresponding triangle.

As mentioned above, we place the only vertex that is connected to vertices of s_l and s_r on I_1, if it exists, and we place the vertices only connected to s_l on c_l and the vertices only connected to s_r on c_r. We further place all vertices without inter-cluster edges on σ in the order of the embedding, possibly in between vertices that are already drawn. As σ is convex, we can then draw all intra-cluster edges of B. Afterwards, we can apply the algorithm for drawing a graph with a prescribed convex outer face of Chambers et al. [4] to complete the drawing of B with the interior of B. Ultimately we draw the inter-cluster edges $E_{\text{inter}}(B)$.

3.2 Correctness of the Construction

To prove the correctness of the resulting drawing Γ, we have to show that it has the following four properties:

(1) Γ is $\mathbb{C}$-*valid.*

(2) Γ is *straight-line.*

(3) Γ has *embedding* $\mathcal{E}$.

(4) Γ is *planar.*

(1) Γ is $\mathbb{C}$-valid: The positions of the vertices $(V \setminus V(B) \setminus V_\circ(B)) \cup \{v_c\}$ in Γ are the same as in Γ'. The drawing Γ' is $\mathbb{C}$-valid. So Γ is $\mathbb{C}$-valid if and only if all vertices $V(B) \cup V_\circ(B)$ lie inside the ϵ-circle of G_i. The vertices $V(B) \cup V_\circ(B)$ are placed inside c or on its border and c lies properly inside the ϵ-circle of G_i (see property (I) of Lemma 3.4).

(2) Γ is straight-line: In Γ', all edges are drawn straight-line and all edges we added in the construction above are drawn straight-line. Therefore, also Γ is straight-line.

(3) Γ has embedding $\mathcal{E}$: The drawing Γ' has the embedding $\mathcal{E}'$, which is implied by $\mathcal{E}$ through the contraction of B. So we only need to show that the vertices $V(B), V_\circ(B)$ and $N_{\text{inter}}(B)$ have the correct embedding in Γ.

If B has no inter-cluster edges and no parent block, the clustered graph C only consists of B and its interior. Then the vertices $V(B)$ are placed on the convex shape c. So, as the algorithm of Chambers et al. works correctly, in this case all vertices are embedded correctly.

Consider the cases where B has incident inter-cluster edges and/or a parent block B_p. The circular order of $N_{\text{inter}}(v_c)$ around every point in c is the same as around v_c in C/B, by property (II) of Lemma 3.4. This is the same as the circular order of $N_{\text{inter}}(B)$ around v_c in C. This means that the ordering of all inter-cluster edges around the vertices $V(B)$ is correct, as we showed in (2) that all edges are drawn straight-line.

The shape σ is drawn inside the correct pie slice of c according to $\mathcal{E}$, so the ordering of the edges around v_c is correct if the interior of B is embedded correctly.
The graph B is a cycle or contains only one edge and it is embedded on the convex shape σ. So the edges $E(B)$ are positioned completely inside σ. As no edge of $E_{\text{inter}}(B)$ crosses σ (see proof of property (4a)) and the algorithm of Chambers et al. positions the vertices

and edges of the interior of B inside the convex polygon formed by B in correct order, all vertices $V(B) \cup V_\circ(B)$ are embedded correctly.

Now we have to show that the embedding of all $N_{\text{inter}}(B)$ is correct. We show in (4) that there are no crossings among the edges $E_{\text{inter}}(B)$ and that all vertices $V(B)$ are placed in such a way that the straight-line connections from them to their inter-cluster neighbours do not cross any components of B. This means that the order in which the vertices $V(B)$ of the cycle B are placed on the circle c is the same order in which they are connected to the vertices $N_{\text{inter}}(B)$. This order is exactly the one that is given by $\mathcal{E}$.
Consider a vertex $u \in N_{\text{inter}}(B)$. Let $E_B(u)$ be the edges of $E_{\text{inter}}(B)$ that are incident to u. What is left to show is that the edges $E_B(u)$ are sorted in correctly in between other edges incident u. As B is free, the edges $E_B(u)$ are embedded consecutively around u. In C/B the vertex u is embedded correctly. This means the edge (u, v_c) is sorted in correctly. When unpacking B, all edges incident to u except of (u, v_c) stay at their positions and (u, v_c) is replaced by $E_B(u)$. Therefore all $N_{\text{inter}}(B)$ are embedded correctly.

(4) Γ is planar: The drawing Γ' is planar. So we need to show that all edges that we add for Γ do not cross the edges that C and C/B have in common. And also that the added edges do not cross each other. This means we have to show the following properties:

(a) No edge of the set $E(B) \cup E_\circ(B)$, crosses any edge of E.

(b) No edge of the set $E_{\text{inter}}(B)$, crosses any edge of E.

Proof of (a) If B has no parent block and no inter-cluster edges, it is the only component of C. Its vertices $V(B)$ are positioned on c. Thus they describe a convex polygon, which means the algorithm of Chambers et al. can be used and produces a planar drawing of B and its interior.

In all other cases the vertices of B are placed on σ. Since σ is strictly convex, all edges of B are placed inside σ. Because the algorithm of Chambers et al. places the vertices $V_\circ(B)$ inside the convex polygon formed by B, all edges $E(B) \cup E_\circ(B)$ are placed inside σ. We constructed σ either between the bordering edges of B or, if there are no bordering edges of B, we placed it between the intra-cluster edges f and g of the parent block B_p. In the latter case we constructed a new circle c' inside c which does only contain parts of f and g and the vertex v_c in Γ'. This means no vertices and edges of G_i, except those of B and its interior, are inside σ. So the edges $E(B) \cup E_\circ(B)$ cannot cross any components of the already drawn graph. Also they do not intersect each other, as the algorithm of Chambers et al. works correctly.
What is left to prove is that the edges $E(B) \cup E_\circ(B)$ do not intersect the edges $E_{\text{inter}}(B)$. So in the following we only need to consider the case in which bordering edges of B exist.

To do this, we show that no edge of $E_{\text{inter}}(B)$ crosses σ. In the following we see that this is guaranteed by the construction of σ and the placement of the vertices on σ.
The shape σ is convex and the partitioning into s_l and s_r is done by l. The line l is constructed as bisector between the bordering edges e_l and e_u or, if the bisector crosses a vertex of $N_{\text{inter}}(B)$, it is very near to the position of a bisector. We claim that this guarantees that no inter-cluster edge $(v, u) \in E_{\text{inter}}(B)$ incident to a vertex v placed on σ crosses l.
The elongations of the two segments $\overline{uv_c}$ and $\overline{vv_c}$ enclose a region r of c that does not include e_l or e_u (see Figure 3.11). This region r encloses an angle α in v_c. When $\alpha \leq 180°$, then r is convex. The line l lies outside of r as v is placed on the according curve of σ. So a crossing of (v, u) with l cannot happen when $\alpha \leq 180°$.
Consider the case where $\alpha > 180°$. As u is connected straight-line to v_c in Γ', it has to lie in the region that is bounded by the extended bordering edges. As l is (nearly) a bisector

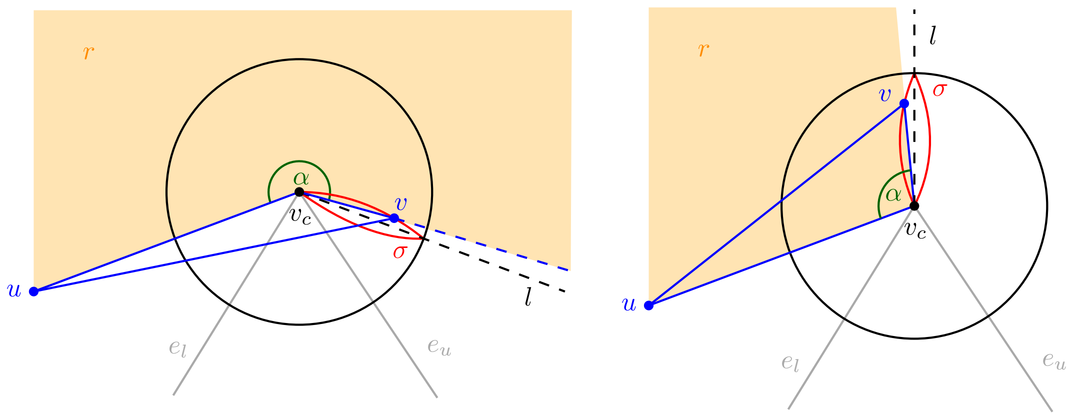

Figure 3.11: On the left $\alpha > 180°$ holds and (v,u) crosses l. On the right l is the bisector between e_l and e_u, so $\alpha < 180°$ holds and (v,u) does not cross l. The region r is, in both cases, marked in orange.

of this region, it partitions the angle enclosed by e_l and e_u into two angles that are both smaller or equal to 180°. So r has to be convex, which means that (v,u) does not cross l.

The fact that l is not crossed by edges of $E_{\text{inter}}(B)$ guarantees that the construction of the triangles $\triangle_l$ and $\triangle_r$ is valid. Therefore the curves c_l and c_r can be constructed correctly.

Now we need to show that the edges $E_{\text{inter}}(B)$ do not cross c_l and c_r. Let S_l be the set of segments connecting v_c and I_1 to all vertices of s_l (equivalently for S_r and s_r). We claim that no segment of S_l crosses $\triangle_l$.
This is ensured by property (II) of c (see Lemma 3.4), which states that the ordering of inter-cluster edges around v_c stays the same when shifting v_c to any point in c. This means that the ordering of inter-cluster edges around v_c is the same as around I_1. So the segments $\overline{I_1u_1}$ and $\overline{v_cu_q}$ are closest to $\overline{v_cI_1}$ and, as $\overline{I_1u_1}$ and $\overline{v_cu_q}$ are the sides of $\triangle_l$, no segment of S_l can cross $\triangle_l$. This also means that all $u \in s_l$ can be connected straight-line to every point of c_l without intersecting c_l. The equivalent statement for $\triangle_r$ and the vertices in s_r is also true. Additionally, this guarantees that a vertex placed on I_1 can be connected straight-line to every vertex in $N_{\text{inter}}(B)$ without intersecting σ.
Eventually we see that no edge of $E(B) \cup E_\circ(B)$ crosses any edge of E.

Proof of (b) Next we consider the added inter-cluster edges $E_{\text{inter}}(B)$. We just showed that they do not cross any edges of $E(B) \cup E_\circ(B)$. An inter-cluster edge $(v,u) \in E_{\text{inter}}(B)$ lies inside cone(u,c). This cone only contains components of G_i and the vertex u, by construction of c (see proof of requirement (II) in Lemma 3.4). So only edges that are incident to G_i can be crossed by (v,u). All vertices $N_{\text{inter}}(B)$ lie in the region r_{inter} bounded by the extended bordering edges of B, as they were connected straight-line to v_c in Γ' (see Figure 3.12). As we examine possible crossings of the edges$E_{\text{inter}}(B)$, the block B has bordering edges. In r_{inter} the shape σ is placed, too. So to exclude crossings of the edges $E_{\text{inter}}(B)$ with components of C that are not B or its interior, it suffices to show that the bordering edges e_l and e_u are not crossed.

When e_l and e_u enclose an angle bigger than 180°, the region they enclose is not convex. Only in this case e_l and e_u can be crossed by inter-cluster edges. Extend e_l and e_u to lines l_l and l_u (see Figure 3.12). The edge (v,u) can only cross e_l or e_u if v is placed in the region r_f of c that is enclosed by e_l and l_u or in the region r_g that is enclosed by e_u an l_l. Suppose that v is placed in r_g. Then, by the construction, it is placed on c_r and therefore only connected to vertices of s_r. Equivalently, vertices placed in r_f are placed on c_l and are only connected to vertices in s_l. So no bordering edge is crossed by edges of $E_{\text{inter}}(B)$.

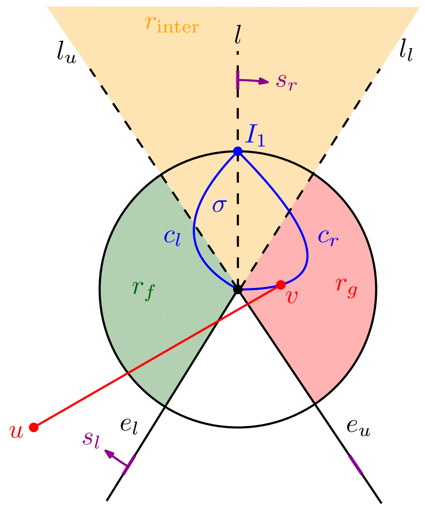

Figure 3.12: The region r_{inter}, inside which all $N_{\text{inter}}(B)$ lie, is highlighted in orange. The region r_f is green and r_g is red. A vertex v on c_r has to be connected to vertices in the sequence s_r. So the edge (v, u) cannot occur.

It only remains to show that the edges $E_{\text{inter}}(B)$ do not cross each other. We saw that σ is not crossed by the edges $E_{\text{inter}}(B)$ and, as the curves are strictly convex, there cannot be any collinear points on them. From the claim we saw that all sets U_v include consecutive sequences of vertices and that two of them can at most have one vertex in common. Also we saw in (3) that Γ has the correct embedding. So the edges $E_{\text{inter}}(B)$ cannot cross each other.

4. Conclusion

Given a planar straight-line drawing of a cluster-graph G_C with an embedding corresponding to the embedding of a simple connected clustered graph $C = (G, \mathcal{V})$, we asked ourselves the following question: Could we "unpack" the drawing of G_C to a planar straight-line drawing Γ of C? Or would we need to abolish the requirement of Γ to be straight-line, while working on the construction? "To bend or not to bend?", that was our question. Then we developed the construction described in this thesis, which proves that it is possible to draw C with all the wanted properties. So the answer to our question is: "Not to bend!"

Our construction works as follows: We create a block-cut tree for each cluster in the skeleton of C. This divides the skeleton of the clusters into blocks. Then, beginning with the cluster-graph G_C, we iteratively unpack these blocks inside of circles around the vertices of the cluster-graph. When unpacking one block, we construct a convex shape in a "suitable" place and then place the vertices of the block on its boundary. The construction of this shape and the placement of vertices on it, is done in a way that guarantees that no edge-crossings occur and that the embedding is not destroyed. Degenerated blocks do not have any interior. But non-degenerated blocks of the skeleton have and they are always simple cycles. They describe a convex polygon, when placed on the boundary of a convex shape. Because of this we can then use the algorithm for drawing a graph with a prescribed convex outer face of Chambers et al. [4] to draw the interior of the blocks.

We restricted the clustered graph to be connected. But the same construction also works for graphs that have connected clusters but are not connected themselves. The only thing one has to be careful about in this case is, that you cannot draw the different components completely independently. This is because the whole circle arrangement with the circles of all components has to be valid.

Our construction has some degrees of freedom. These might be useful to improve the resulting drawing. One can vary how to choose the root blocks of the block-cut tree. Choose, for example, one that minimizes the height of the tree. Also, in different steps of the algorithm, it is not completely fixed where the vertices have to be placed. One can vary this for example by keeping a big distance to neighbouring vertices to make the drawing easier to read.

In the following we propose two further questions related to our construction. The first proposes a way to improve the layout of the constructed drawing and the second is a proposal for further generalisation of the problem.

Feng et al. [6] show that there exist clustered graphs that require exponential area for any straight-line drawing in which clusters are bordered by convex regions. In our construction the clusters are bordered by circles. Therefore the calculated drawing has to require exponential area. But even disregarding this, our algorithm will possibly not compute very "nice" drawings of G. This is because the area where a block can be unpacked is restricted very much by the convex shape that we construct. In many cases this restriction could be relaxed. One possibility to make the drawing "nicer" afterwards might be to use our algorithm to compute the positions of the vertices and, after that, use a force directed algorithm to let the vertices spread around. To do this the forces have to be designed in such a way that the planarity and the embedding of the drawing is kept. Also there should be forces that keep the vertices inside or near to their ϵ-circles.

We required the clustered graph C to be simple, as, otherwise, no $\mathbb{C}$-valid planar straight-line drawing with any circle arrangement $\mathbb{C}$ with respect to C would be possible. Is there a way to adapt the construction for non-simple clustered graphs C? For this, one would need to adapt the circle arrangement, e.g. allow circles of different sizes. Also there might be no planar straight-line drawing of G_C, in this case. So one would have to find an algorithm that gets a drawing of G_C as input that is not straight-line. What would be the constraints for Γ to make it look similar to the drawing of G_C? Would one want to bend edges by purpose? Or would one only allow to vary the sizes of the circles of $\mathbb{C}$?

Bibliography

[1] Md. Jawaherul Alam, Michael Kaufmann, Stephen G. Kobourov, and Tamara Mchedlidze. Fitting planar graphs on planar maps. *Journal of Graph Algorithms and Applications*, 19(1):413–440, 2015.

[2] Patrizio Angelini, Fabrizio Frati, and Michael Kaufmann. Straight-line rectangular drawings of clustered graphs. *Discrete & Computational Geometry*, 45(1):88–140, 2011.

[3] Ralf Brockenauer and Sabine Cornelsen. *Drawing Clusters and Hierarchies*, pages 193–227. In Kaufmann and Wagner [9], 2001.

[4] Erin W. Chambers, David Eppstein, Michael T. Goodrich, and Maarten Löffler. Drawing graphs in the plane with a prescribed outer face and polynomial area. *Journal of Graph Algorithms and Applications*, 16(2):243–259, 2012.

[5] Peter Eades and Qing-Wen Feng. *Multilevel visualization of clustered graphs*, pages 101–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[6] Qing-Wen Feng, Robert F. Cohen, and Peter Eades. How to draw a planar clustered graph. In *Computing and Combinatorics, First Annual International Conference, COCOON '95, Xi'an, China, 1995, Proceedings*, pages 21–30, 1995.

[7] Qing-Wen Feng, Robert F. Cohen, and Peter Eades. Planarity for clustered graphs. In Paul Spirakis, editor, *Algorithms - ESA '95, Third Annual European Symposium, Corfu, Greece, 1995, Proceedings*, pages 213–226. Springer Berlin Heidelberg, 1995.

[8] István Fáry. On straight-line representation of planar graphs. In *Acta Scientiarium Mathematicarum 11*, pages 229–233. János Bolyai Mathematical Institute (University of Szeged), 1948.

[9] Michael Kaufmann and Dorothea Wagner, editors. *Drawing Graphs Methods and Models*, volume 2025 of *Lecture Notes in Computer Science*. Springer, 2001.

[10] Marcus Schaefer. *Toward a Theory of Planarity: Hanani-Tutte and Planarity Variants*, pages 162–173. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[11] René Weiskircher. *Drawing Planar Graphs*, pages 23–45. In Kaufmann and Wagner [9], 2001.