Università degli Studi di Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

Clustering Cycles into Cycles of Clusters

P. F. Cortese, G. Di Battista, M. Patrignani, and M. Pizzonia

RT-DIA-91-2004 **September 2004**

Dipartimento di Informatica e Automazione,
Università di Roma Tre,
Rome, Italy.
{cortese,gdb,patrigna,pizzonia}@dia.uniroma3.it

Work partially supported by European Commission - Fet Open project COSIN - COevolution and Self-organisation In dynamical Networks - IST-2001-33555, by European Commission - Fet Open project DELIS - Dynamically Evolving Large Scale Information Systems - Contract no 001907, by "Project ALGO-NEXT: Algorithms for the Next Generation Internet and Web: Methodologies, Design, and Experiments", MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale, and by "The Multichannel Adaptive Information Systems (MAIS) Project", MIUR-FIRB.

ABSTRACT

In this paper we study simple families of clustered graphs that are highly non connected. We start by studying 3-cluster cycles, that are clustered graphs such that the underlying graph is a simple cycle and there are three clusters all at the same level. We show that in this case testing the c-planarity can be done efficiently and give an efficient drawing algorithm. Also, we characterize 3-cluster cycles in terms of formal grammars. Finally, we generalize the results on 3-cluster cycles considering clustered graphs that have a cycle structure at each level of the inclusion tree. Even in this case we show efficient c-planarity testing and drawing algorithms

1 Introduction

Given a graph, a *cluster* is a non empty subset of its vertices. A *clustered graph* $C(G, T)$ consists of a graph G and a rooted tree T such that the leaves of T are the vertices of G (see Fig. 1 for an example). Each internal node ν of T corresponds to the cluster $V(\nu)$ of G whose vertices are the leaves of the subtree rooted at ν. The subgraph of G induced by $V(\nu)$ is denoted as $G(\nu)$. An edge e between a vertex of $V(\nu)$ and a vertex of $V - V(\nu)$ is said to be *incident* on ν. Graph G and tree T are called *underlying graph* and *inclusion tree*, respectively.

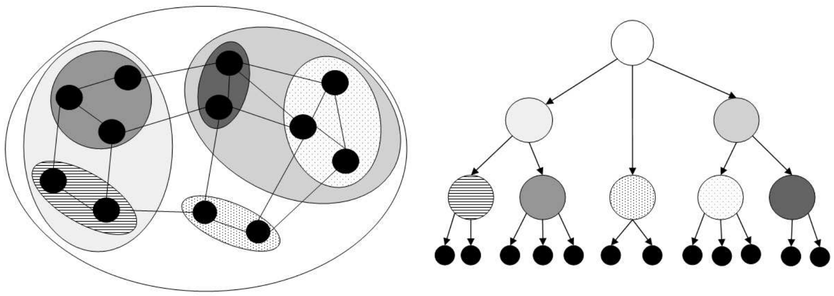


Figure 1: An example of a clustered graph and its inclusion tree.

Several applications may benefit from drawing clustered graphs. In many areas, as, for example, networking and telecommunications, the vertices of the graphs are naturally partitioned into clusters, and the representation of such clusters, which is generally required by the applications, may be also of help in the challenge of exploring large amounts of data [14]. Conversely, clusters may be artificially introduced with the purpose of navigating large graphs, allowing representations at different levels of abstraction (see for example [7]).

In a *drawing* of a clustered graph $C(G, T)$ vertices and edges of G are drawn as points and curves as usual [5], and each node ν of T is a simple closed region $R(\nu)$ such that: (i) $R(\nu)$ contains the drawing of $G(\nu)$; and (ii) $R(\nu)$ contains a region $R(\mu)$ if and only if μ is a descendant of ν in T.

Consider an edge e and a node ν of T. If e is incident on ν and e crosses the boundary of $R(\nu)$ more than once, we say that edge e and region $R(\nu)$ have an *edge-region crossing*. Also, edge e and region $R(\nu)$ have an *edge-region crossing* if e is not incident on ν and e crosses the boundary of $R(\nu)$. A drawing of a clustered graph is *c-planar* if it does not have edge crossings and edge-region crossings. A clustered graph is *c-planar* if it admits a c-planar drawing. The concept of c-planarity extends that of planarity, which is a recognized aesthetic criterion for graph drawings.

Clustered planarity, because of its practical impact and because of its theoretical appeal, attracted many research contributions.

Feng, Cohen, and Eades devised the first polynomial time c-planarity testing algorithm for connected clustered graphs [13, 12]. A clustered graph is *connected* if for each node ν of T we have that $G(\nu)$ is connected (for example, the clustered graph in Fig. 1 is connected). Based on the above result, multilevel, straight line, or orthogonal drawings of connected c-planar graphs can be obtained by using the techniques described in [8, 9, 10], respectively. A planarization algorithm for (non c-planar) connected clustered graph is shown in [4].

However, the complexity of the c-planarity testing for a non connected clustered graph is still unknown.

A contribution on this topic has been given by Gutwenger et al. that presented a polynomial time algorithm for c-planarity testing for *almost connected* clustered graphs [15]. In almost connected clustered graphs either all nodes corresponding to non connected clusters are in the same path in T starting at the root of T, or for each non connected cluster its parent and all its siblings are connected. Note that the set of almost connected clustered graphs contains that of connected clustered graphs (see Fig. 2). Another contribution studying the interplay between c-planarity and connectivity has been presented in [3] by Cornelsen and Wagner. They show that a *completely connected* clustered graph is c-planar iff its underlying graph is planar. A completely connected clustered graph is so that not only each cluster is connected but also its complement is connected.

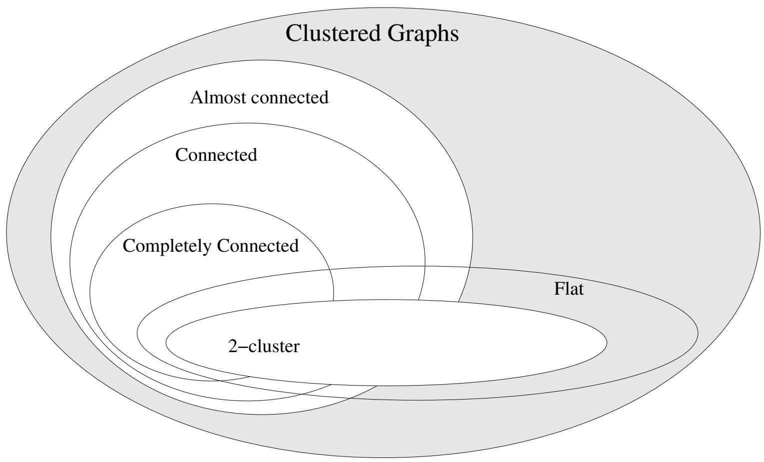


Figure 2: Classes of clustered graphs.

With the purpose of investigating the complexity of c-planarity testing in the general case, we focus on simple classes of non connected clustered graphs. We say that a clustered graph $C(G, T)$ is *flat* if T has depth two, that is, if all the clusters are at the same level with the exception of the root of T. In such a case, a clustered graph is equivalently described by a graph G and a labeling of its nodes with labels $\{l_1, l_2, \ldots, l_k\}$, where k is the number of clusters at the lower level. We say that a flat clustered graph is a *k-cluster graph* if such a labeling uses k different labels.

For 2-cluster graphs, the results in [2, 1] by Biedl, Kaufmann, and Mutzel can be interpreted as a linear time c-planarity test for non connected clustered graphs. A similar result can be obtained from the work of Di Battista, Liu and Rival [6], where it is shown that a planar bipartite graph always admits an upward drawing.

As far as we know, the complexity of testing whether a 3-cluster graph is c-planar is still open. In this paper we restrict our attention to 3-cluster graphs whose underlying graph is a cycle. Discovering the complexity of finding a c-planar drawing for 3-cluster cycles may help investigating the c-planarity problem for 3-cluster graphs.

Fig. 3.a provides an example of a 3-cluster cycle. As it is shown in the following sections, the problem of finding a c-planar drawing for the 3-cluster cycle of Fig. 3.a is equivalent to the problem to add new edges so that: (i) the new graph (i.e. cycle plus

new edges) is planar and (ii) for each label, the subgraph induced by the vertices with that label is connected. In the case of Fig. 3.a the problem admits a solution, depicted in Fig. 3.b. The set of edges added to the cycle, that we call *saturator*, is used to "simulate" the closed regions containing the clusters (See Fig. 3.c). Observe that the clustered graph of Fig. 3 is non connected.

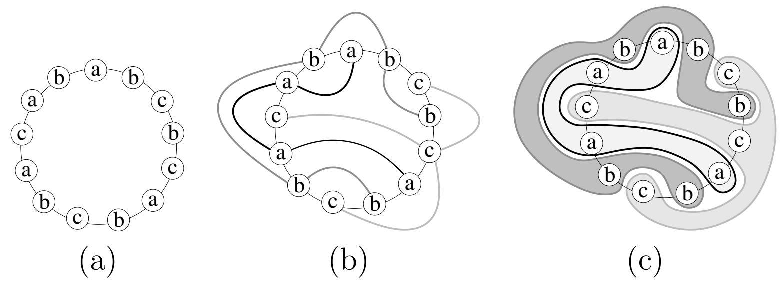


Figure 3: (a) An example of a cycle with labels in $\{a, b, c\}$. (b) The cycle with extra edges. (c) The corresponding clustered drawing of the cycle.

In this paper we present the following results.

- In Section 3 we study 3-cluster cycles. We show that in this case testing the c-planarity can be done efficiently. We also give an efficient drawing algorithm. Further, we show that in this specific case, if the c-planarity problem admits a solution then a saturator exists that is composed only by simple paths.
- In Section 4 we characterize 3-cluster cycles in terms of formal grammars.
- Finally, in Section 5 we generalize the results on 3-cluster cycles considering clustered graphs that have a cycle structure at each level of the inclusion tree. Even in this case we show efficient c-planarity testing and drawing algorithms.

Section 2 contains preliminaries, while conclusions and open problems are in Section 6.

2 Preliminaries

We assume familiarity with connectivity and planarity of graphs [11, 5]. We also assume familiarity with formal grammars [16].

Given a non connected clustered graph $C(G, T)$, a *saturator* of C is a set of edges that can be added to the underlying graph G so that C becomes connected. If G with the added edges is c-planar, we say that the saturator is *planar*, *non-planar* otherwise. It is easy to see that a non-connected clustered graph is c-planar iff it admits a planar saturator. Finding a saturator of a clustered graph is important since it allows to apply to C the same drawing techniques that have been devised for connected clustered graphs.

We call *3-cluster cycle* a clustered graph such that the underlying graph is a simple cycle and there are exactly three clusters all at the same level (plus the root cluster). In a 3-cluster cycle the inclusion tree consists of a root node with three children. Each vertex of the underlying cycle is a child of one of these three nodes. Given a 3-cluster cycle, we associate a label in $\{a, b, c\}$ to each of the three clusters. Observe that there

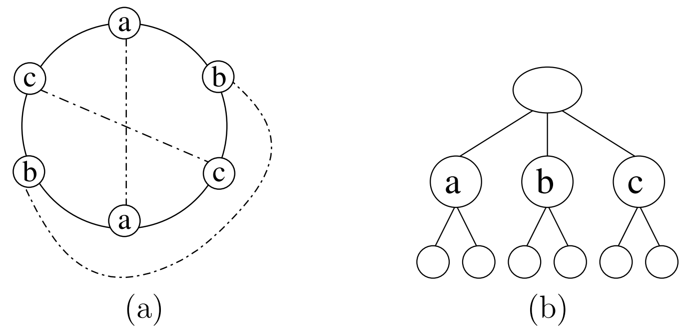


Figure 4: The smallest 3-cluster cycle that is not c-planar. (a) The cycle with labels; the dashed lines represent the unique saturator. Note that the cycle and its saturator form a $K_{3,3}$ graph, and thus the saturator is not planar.(b) The corresponding inclusion tree.

exist 3-cluster cycles that are not c-planar. Fig. 4 provides an example of a non c-planar 3-cluster cycle.

Consider a 3-cluster cycle and arbitrarily select a starting vertex and a direction. We can visit the cycle and denote it by the sequence σ of labels associated with the clusters encountered during the visit. The same 3-cluster cycle is also denoted by any cyclic permutation of σ and by any reverse sequence of such permutations. We use Greek letters to denote general sequences and Roman letters to identify single-character sequences. Given a sequence σ, we denote with $\overline{\sigma}$ its reverse sequence.

It is easy to see that repeated consecutive labels can be collapsed into a single label without affecting the c-planarity property of a 3-cluster cycle. Hence, in the following we consider only 3-cluster cycles where consecutive vertices belong to distinct clusters. Also, since clusters can not be empty, in a 3-cluster cycle at least one occurrence of each label can be found.

We assign a cyclic order to the labels a, b, c so that $a \prec b$, $b \prec c$, and $c \prec a$. A sequence σ is *monotonic increasing* (*decreasing*) if for each pair x, y of consecutive labels of σ $x \prec y$ ($y \prec x$). A sequence is *cyclically increasing (decreasing) monotonic* if all its cyclic permutations are increasing (decreasing) monotonic.

Given a 3-cluster cycle σ, $Balance(\sigma)$ is a number defined as follows. Select a start vertex and a direction. Set counter c to zero. Visit σ adding (subtracting) one unit to c when passing from x to y, where $x \prec y$ ($y \prec x$). Observe that, when the start vertex is reached again, c is a multiple of 3 that can be positive, negative, or zero. If we selected a different start vertex, while preserving the direction, we would obtain the same value. On the contrary, if σ was visited in the opposite direction the opposite value would be obtained for c. $Balance(\sigma) = |c|$. For example, $Balance(ababc) = 3$ and $Balance(cbacba) = 6$.

Observe that, when representing a 3-cluster cycle with a sequence of labels, by reading the sequence from left to right, we implicitly choose a direction for visiting the cycle. For simplicity, we adopt the convention of representing a 3-cluster cycle with a sequence σ such that, when the vertices of the cycle are visited according to the order induced by σ, a non negative value for c is obtained.

3 Cycles with Three Clusters

In this section we address the problem of testing the c-planarity of a 3-cluster cycle. Lemma 1 introduces a transformation, called “zig-zag removal”, that can be applied to a 3-cluster cycle, to obtain a smaller 3-cluster cycle which is c-planar if and only if the starting 3-cluster cycle is c-planar. Lemma 2, by repeatedly applying the above mentioned tranformation, shows that any 3-cluster cycle can be reduced to a trivial 3-cluster cycle which is either cyclically monotonic or it is composed by two maximal monotonic subsequences. Lemma 3 and Lemma 4 show how to test c-planarity for these two kinds of 3-cluster cycles. Finally, Theorems 1, 2, and 3 state the main results about c-planarity testing of 3-cluster cycles.

Lemma 1 ***(Zig-zag removal)*** *Let $\sigma = \sigma_1 x \alpha y \overline{\alpha} x \alpha y \sigma_2$ be a 3-cluster cycle such that σ_1, σ_2, and α are possibly empty and $x\alpha y$ is monotonic. The 3-cluster cycle $\tau = \sigma_1 x \alpha y \sigma_2$ is c-planar if and only if σ is c-planar. Balance(σ) = Balance(τ).*

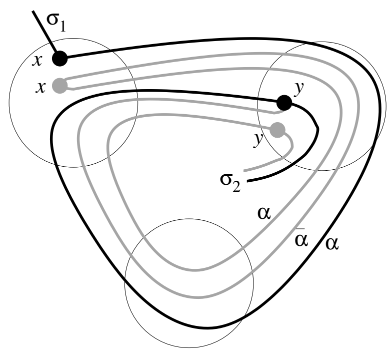


Figure 5: Illustration of the proof of Lemma 1. Circles represent clusters.

Before showing a formal proof for this Lemma, we give an intuitive description of the main idea used for the proof.

Suppose there exists a c-planar drawing of τ. The black line in Fig. 5 shows an example of such a drawing for the portion concerning subsequence $x\alpha y$. Introduce between y and the first vertex of σ_2 the sequence $\overline{\alpha} x \alpha y$, drawing it suitably close to $x\alpha y$ so to preserve c-planarity. The result is shown in Fig. 5 where the added part is drawn gray.

Conversely, suppose that there exists a c-planar drawing of σ. Fig. 6.a shows an example of such a drawing for the part concerning subsequence $x\alpha y\overline{\alpha} x\alpha y$. The inlet formed by $x\alpha y\overline{\alpha} x$ may contain parts of σ that are denoted by Q in Fig. 6.a. Analogously, the parts of σ that are contained in the inlet formed by $y\overline{\alpha} x\alpha y$ are denoted by P. The embedding of P and Q may be rearranged preserving c-planarity as shown in Fig. 6.b. Path $\overline{\alpha} x\alpha y$ can now be replaced with an edge connecting vertex y with the first vertex of σ_2.

A formal proof for Lemma 1 follows.

Proof: In the following we use primes to distinguish different occurrences of the same character (as in x' and x'') or of the same substring (as in α' and α''). According to this notation $\sigma = \sigma_1 x' \alpha' y' \overline{\alpha} x'' \alpha'' y'' \sigma_2$ and $\tau = \sigma_1 x' \alpha' y' \sigma_2$.

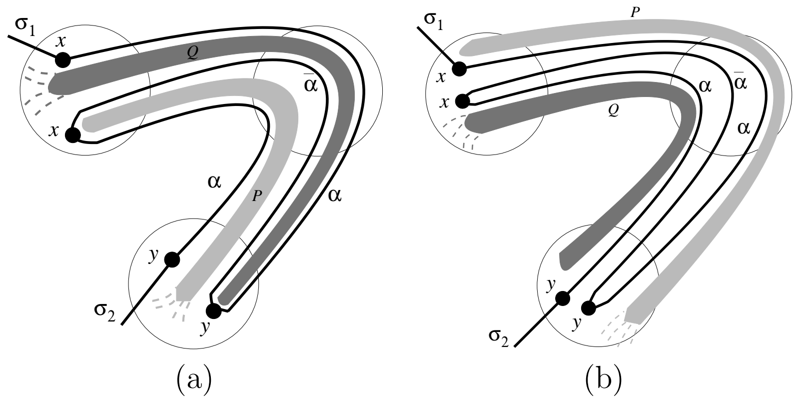


Figure 6: Zig-zag removal. (a) Starting configuration. (b) Rearrangement of the embedding.

We denote by k the length of the sequences α', $\overline{\alpha}$, and α''. We denote with $\alpha(j)$ the j-th vertex of sequence α where $j \in \{1, \ldots, k\}$. Suppose that there exists a c-planar drawing Γ_τ of τ (Fig. 7.a shows an example of such a drawing for the part related to the subsequence $x'\alpha'y'$), we prove that σ is c-planar by constructing a c-planar drawing Γ_σ in the following way.

All vertices and clusters boundaries of Γ_τ are drawn in Γ_σ in the same way as in Γ_τ. Edges of Γ_τ are drawn in Γ_σ in the same way as in Γ_τ with the exception of the edge between y' and the vertex z following y' in τ. Such an edge is replaced by the path $\overline{\alpha}x''\alpha''y''$ with $\overline{\alpha}$ connected to y' and y' connected to z. This path is drawn as follows. Vertex x'' is placed arbitrarily near to x' in the region of Γ_σ that is on the left side of the edge joining x' with $\alpha'(1)$. Vertex y'' is placed arbitrarily near to y' in the region of Γ_σ that is on the left side of the edge joining y' with $\alpha'(k)$. For each vertex $\alpha'(j)$ the

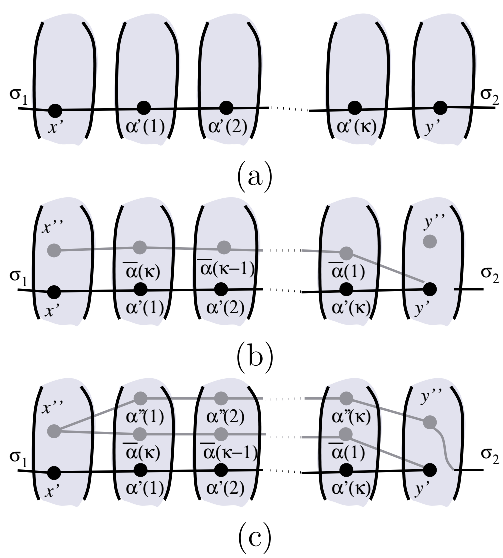


Figure 7: (a) The drawing of $x'\alpha'y'$ in Γ_τ. The gray zones are part of cluster regions. Note that τ may pass through each cluster many times. (b) After the insertion of path $y'\overline{\alpha}x''$. (c) After the insertion of path $x''\alpha''y''$, the last edge is connected to the first vertex of σ_2.

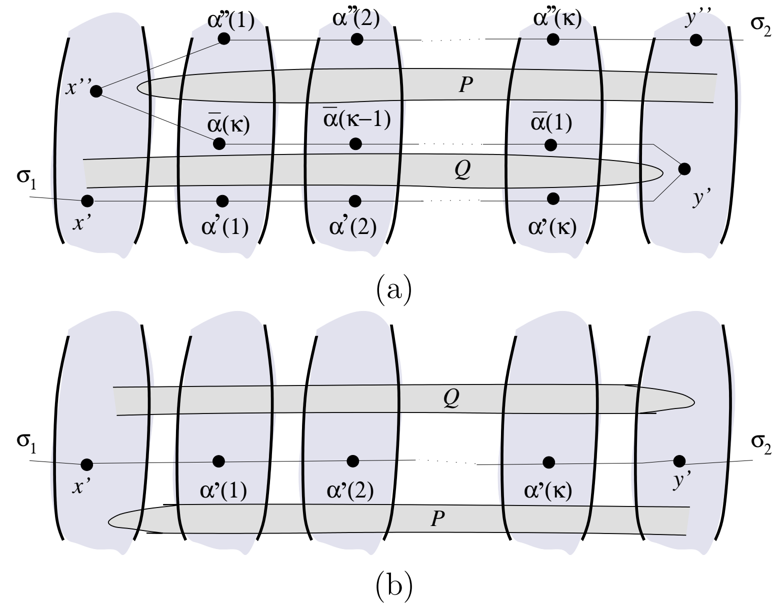


Figure 8: (a) The drawing of $x'\alpha'y'\overline{\alpha}x''\alpha''y''$ in Γ_σ. The gray zones are part of cluster regions. Note that τ may pass through each cluster many times. (b) The drawing of Γ_τ after $y'\overline{\alpha}x''\alpha''y''$ was deleted and P was moved

corresponding vertex $\overline{\alpha}(k-j+1)$ is placed arbitrarily near to $\alpha'(j)$ into the region that is on the left side when traversing α' from x' to y'. The path $y'\overline{\alpha}x''$ can now be connected without crossings since the edges of $\overline{\alpha}$ can be drawn arbitrarily close to the edges of α' (see Fig. 7.b). Analogously, for each vertex of $\overline{\alpha}(j)$ the corresponding vertex of $\alpha''(k-j+1)$ is placed arbitrarily near to $\overline{\alpha}(j)$ into the region that is on the right side when traversing $\overline{\alpha}$ from y' to x''. The path $x''\alpha''y''$ can now be connected without crossings since edges of α'' can be drawn arbitrarily close to the edges of $\overline{\alpha}$ (see Fig. 7.c). Finally, vertex y'' can be connected to z, crossing the boundary of cluster y only. The drawing Γ_σ is c-planar, since the set of edges added to Γ_τ do not generate edge crossings or edge region crossings.

Suppose σ is c-planar and Γ_σ is a c-planar drawing of it. We prove that τ is c-planar by showing how to build a c-planar drawing Γ_τ starting from Γ_σ.

Call A_1 the set of edges of $x'\alpha'y'$, $\overline{A}$ the set of edges of $y'\overline{\alpha}x''$, and A_2 the set of edges of $x''\alpha''y''$. In Γ_σ (see Fig. 8.a) the inlet delimited by $y'\overline{\alpha}x''\alpha''y''$ and by the part of the border of cluster y between the edges $y'\overline{\alpha}(k)$ and $\alpha''(k)y''$ may contain some parts of σ. Call P the set of edges of such parts including those that cross the border of the cluster of y'. Analogously, consider the inlet delimited by $x'\alpha'y'\overline{\alpha}x''$ and by the part of the border of the cluster of x' between the edges $x'\alpha'(1)$ and $\overline{\alpha}(1)x''$. This inlet may contain some parts of σ. Call Q the edges of the subgraph in this inlet including those that cross the boundary of the cluster of x'.

To construct Γ_τ we delete the path $\overline{\alpha}x''\alpha''y''$. Then, we move the subgraph induced by P, which is on the left side of $x'\alpha'y'$, to the right side of $x'\alpha'y'$ (see Fig. 8.b). This operation may be performed without introducing intersections by suitably shrinking the graph induced by P and placing it close to α'. Note that at this point the subgraph induced by P and the subgraph induced by Q are on the opposite sides of $x'\alpha'y'$, and that y' can be connected to the vertex z following y' in τ crossing the boundary of the cluster of y' only.

Finally, observe that, since we have removed from σ two monotonic sub-sequences, one increasing and one decreasing, with the same length, $Balance(\tau) = Balance(\sigma)$. $\square$

Lemma 1 allows, for example, to study the c-planarity of *cabcab* instead of the c-planarity of *cabcacbabcab* (by taking $\sigma_1 = c$, $x = a$, $\alpha = bc$, $y = a$, and $\sigma_2 = b$).

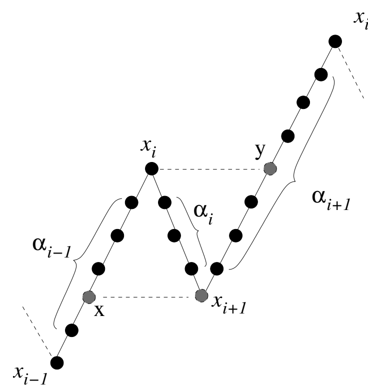


Figure 9: Illustration for the proof of Lemma 2.

Lemma 2 *Let σ be a 3-cluster cycle. There exists a 3-cluster cycle τ such that: $Balance(\tau) = Balance(\sigma)$, τ is c-planar iff σ is c-planar, and either τ is cyclically monotonic or $\tau = x\alpha y\beta$, where*

1. *α and β are non empty,*
2. *$x\alpha y$ is maximal monotonic increasing, and*
3. *$y\beta x$ is maximal monotonic decreasing.*

Proof: If σ is cyclically monotonic the statement is trivially true. If σ is monotonic but not cyclically monotonic note that the length of σ is at least 4. Suppose $\sigma = x_1x_2\alpha_1x_3x_4$, with α_1 possibly empty, and suppose without loss of generality that σ is monotonic increasing (otherwise $\overline{\sigma}$ can be considered). Since all the subsequences of σ are monotonic increasing and $x_4x_1x_2\alpha_1x_3$ is not monotonic, it follows that x_4x_1 is monotonic decreasing. Thus, Lemma 1 can be applied to $x_3x_4x_1x_2\alpha_1$, where σ_1 and α are empty, $x = x_3 = x_1$, $y = x_4 = x_2$, and $\sigma_2 = \alpha'$, obtaining the cycle $x_3x_4\alpha' = x_1x_2\alpha'$, which is cyclically monotonic increasing and which is c-planar iff σ is c-planar.

Otherwise, suppose σ is not monotonic. σ is composed by $m \geq 2$ maximal monotonic sub-sequences. Namely, let $\sigma = x_1\alpha_1x_2\alpha_2 \dots x_m\alpha_m$, where $x_i\alpha_ix_{i+1}$ is maximal monotonic and α_i is possibly empty ($x_{m+1} = x_1$). If $m = 2$, then, since σ is not monotonic, both α_1 and α_2 are non empty and the statement of the lemma holds. If $m > 2$, by applying Lemma 1 we prove that there exists a sequence composed by $m - 2$ maximal monotonic subsequences that is c-planar iff σ is c-planar. By repeatedly applying this argument we find a sequence composed by one or two maximal monotonic subsequences for which one of the cases discussed above applies.

In order to reduce the number m of maximal monotonic sub-sequences by applying Lemma 1, assume that α_i is one of the shortest of such sub-sequences (see Fig. 9) and consider the sub-sequence $x_{i-1}\alpha_{i-1}x_i\alpha_ix_{i+1}\alpha_{i+1}x_{i+2}$. Observe that, it is possible to find in $x_{i-1}\alpha_{i-1}$ an x and in $\alpha_{i+1}x_{i+2}$ a y, such that $x = x_{i+1}$, $y = x_i$, and Lemma 1 can be applied where $x' = x$, $y' = x_i$, $x'' = x_{i+1}$, $y'' = y$, α is the sequence of labels encountered traveling from x to x_i (end vertices excluded), and $\overline{\alpha} = \alpha_i$ (see Fig. 9). □

The following two lemmas (Lemma 3 and Lemma 4) study the c-planarity of the simple families of 3-cluster cycles cited in Lemma 2.

Lemma 3 *A 3-cluster cycle σ such that σ is cyclically monotonic is c-planar if and only if $Balance(\sigma) = 3$.*

Proof: Since σ is monotonic we have that $Balance(\sigma) \neq 0$. Recall that $Balance(\sigma)$ is a multiple of 3. If $Balance(\sigma) = 3$, then it can only be the case that $\sigma = abc$ or $\sigma = bca$ or $\sigma = cab$ and it is trivial to see that σ is c-planar.

Suppose that $Balance(\sigma) \geq 6$. We show that σ is not c-planar. Suppose by contradiction that there exists a c-planar drawing Γ_σ of σ. Consider the first 6 vertices v_1, v_2, v_3, v_4, v_5 and v_6 of σ as drawn in Γ_σ (see Fig. 10). The two edges incident to v_4 separate v_1 from the rest of the vertices of its cluster. Thus, it is possible to add an edge (v_1, v_4) preserving the planarity of the drawing. For similar reasons, it is possible to add the edges (v_2, v_5) and (v_3, v_6). A contradiction arises from the fact that a subdivision of a $K_{3,3}$ can be found in the drawing. Each vertex in $\{v_1, v_3, v_5\}$ is connected to all vertices in $\{v_2, v_4, v_6\}$. Vertex v_1 is connected to v_6 with a path in σ and it is directly connected to v_2 and v_4. Vertices v_3 and v_5 are directly connected to v_2, v_4, and v_6. □

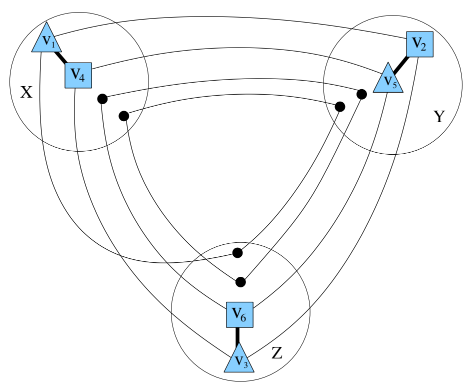


Figure 10: Illustration for the proof of Lemma 3. Triangular and square vertices show the subdivisiono of $K_{3,3}$.

Lemma 4 *Let $\sigma = x\alpha y\beta$ be a 3-cluster cycle, where α and β are non empty, $x\alpha y$ is maximal monotonic increasing, and $y\beta x$ is maximal monotonic decreasing. We have that σ is c-planar iff $Balance(\sigma)$ is in $\{0, 3\}$.*

Proof: Let $Balance(\sigma) = 3k$, with k non negative integer. Suppose k is equal to 0 or 1. A c-planar drawing of σ can be constructed by placing the vertices on three half-lines as in the examples shown in Fig. 11.a and 11.b, respectively. The vertices of each half-line can be enclosed into a region representing their cluster.

Suppose that $k > 1$. We show that σ is not c-planar. Suppose for a contradiction that σ is c-planar and let Γ_σ be a c-planar drawing of σ. Denote with $v_1, \ldots, v_n$ the vertices

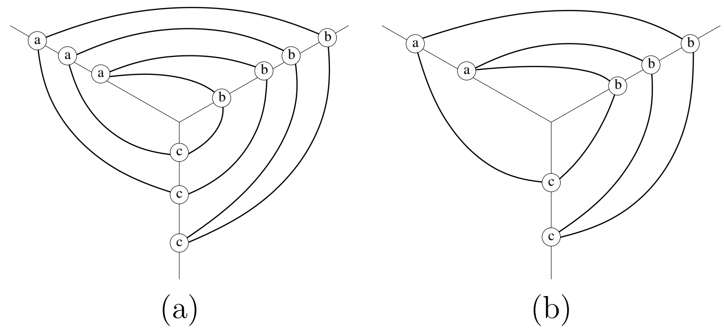


Figure 11: The construction of a c-planar drawing for a cycle σ when $Balance(\sigma) = 0$ (a) and when $Balance(\sigma) = 3$ (b).

of σ starting from the first vertex of α and suppose, without loss of generality, that the length of α is greater or equal than the length of β.

Consider the relative position of v_1 and v_4 in Γ_σ in their cluster X (see Fig. 12). We have that the two edges incident on v_4 separate v_1 from the rest of the vertices of X. Thus, it is possible to join v_1 and v_4 with an edge (v_1, v_4) that is entirely contained into the cluster X and that preserves the planarity of the drawing. Analogously, it is possible to join vertices v_2 and v_5 in cluster Y with the edge (v_2, v_5) and vertices v_3 and v_6 in cluster Z with the edge (v_3, v_6).

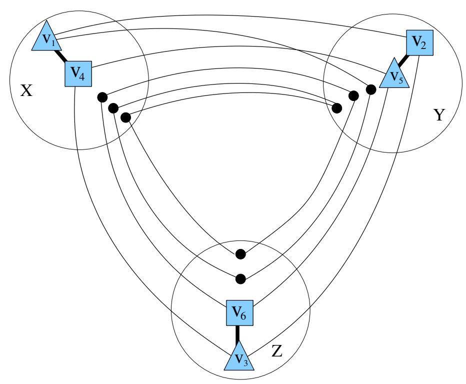


Figure 12: Illustration for the proof of Lemma 4. Triangular and square vertices show the subdivisiono of $K_{3,3}$.

A contradiction arises since a subgraph that is a subdivision of $K_{3,3}$ can be found in the drawing. In fact, exploiting the edges of σ and the edges introduced above, each vertex in $\{v_1, v_3, v_5\}$ is connected to all vertices in $\{v_2, v_4, v_6\}$. Vertex v_1 is directly connected to v_2 and to v_6 with edges of σ, while it is connected to v_4 with edge (v_1, v_4); vertex v_3 is directly connected to v_2 and to v_4 with edges of σ, while it is connected to v_n with edge (v_3, v_n); finally, vertex v_5 is directly connected to v_4 with an edge of σ, it is connected to v_n with a path in σ, and it is connected to v_2 with edge (v_2, v_5). □

Because of Lemma 2, Lemma 3, and Lemma 4, the problem of testing whether a 3-cluster cycle σ is c-planar can be reduced to the problem of computing $Balance(\sigma)$. Since it is easy to compute $Balance(\sigma)$ in linear time (see Section 2), the following theorem holds.

Theorem 1 *Given an n-vertex 3-cluster cycle, there exists an algorithm to test if it is c-planar in $O(n)$ time.*

In what follows we introduce a simple algorithm which guarantees the computation of a c-planar drawing of a 3-cluster cycle, if it admits one, in linear time. Consider a 3-cluster cycle σ with $Balance(\sigma) \in \{0, 3\}$. Set a counter to zero. Visit σ starting from the first vertex and adding (subtracting) one unit to the counter when passing from x to y, where $x \prec y$ ($y \prec x$). Without loss of generality we will assume that the counter never reaches a negative value. Otherwise, we can replace σ with an equivalent cyclic permutation of it that has the above property and that can be obtained in linear time. Let K be the maximum value assumed by the counter during the visit.

We say that a vertex of σ *belongs to the k-th level* iff the counter has value k when reaching such a vertex. The first vertex of σ belongs to level 0. Note that each level contains vertices of the same cluster. Also, vertices belonging to level k and level $k+3$ belong to the same cluster. We denote with $\sigma|_k$ the sequence σ *restricted to level k*, obtained from σ by deleting all the vertices not belonging to the k-th level.

We construct a planar saturator in the following way. For each level $k \in \{0, \ldots, K\}$, we connect with an edge each pair of consecutive vertices of $\sigma|_k$. For each level $k \in \{0, \ldots, K-3\}$, we insert an edge connecting the first vertex of $\sigma|_k$ with the last vertex of $\sigma|_{k+3}$.

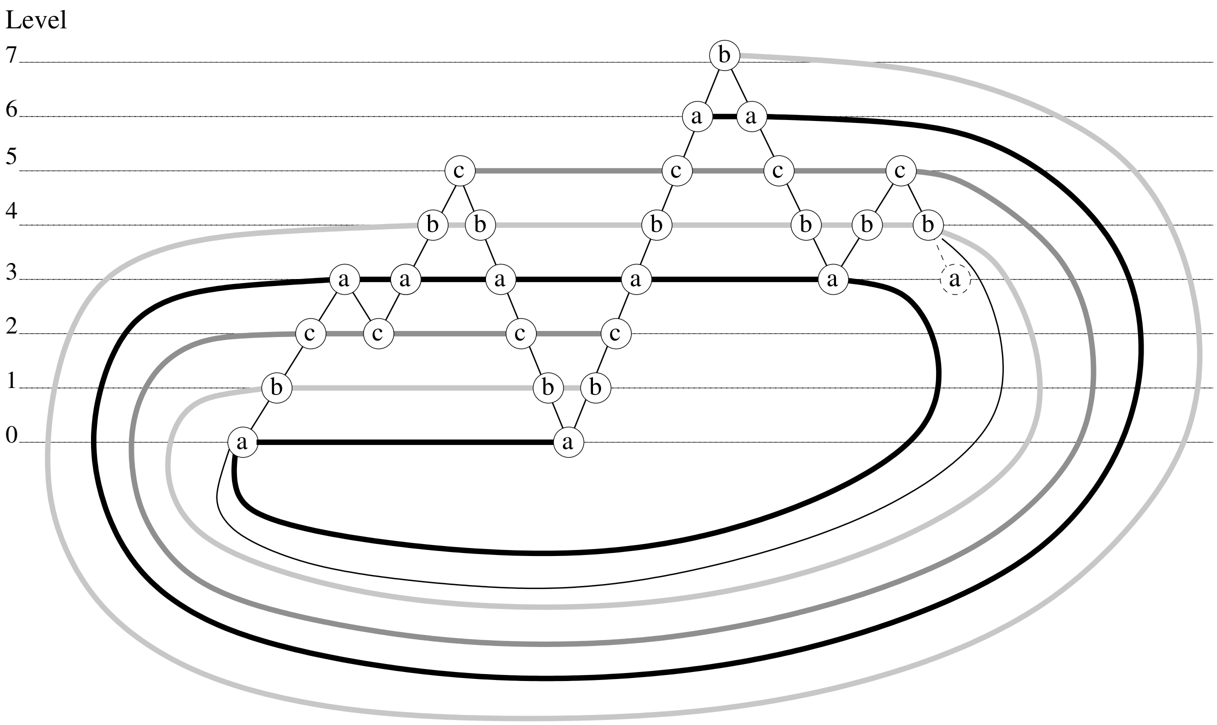


Figure 13: The construction of a c-planar drawing of a 3-cluster cycle σ in the case in which $Balance(\sigma) = 3$.

In order to show that the above defined saturator is planar we provide a c-planar drawing of the graph composed by the cycle and the saturator (see Fig. 13). First, we arrange all the vertices of σ on a grid: the x-coordinate of a vertex is its position in σ and the y-coordinate is its level. Then, we draw each edge of the cycle (excluding the one connecting the first and the last vertex of σ) with a straight segment without introducing intersections. Second, for each level $k \in \{0, \ldots, K\}$, we draw those edges of the saturator that connect pairs of consecutive vertices of $\sigma|_k$ with straight segments without introducing intersections. Note that, the sequence of the clusters at levels $0, \ldots, K-3$ is the same as the sequence of the clusters at levels $3, \ldots, K$. Also, note that at this point of the construction, for each $k \in \{0, \ldots, K\}$ the first and the last vertices of $\sigma|_k$ are on the external face. Hence, the drawing can be completed without intersections by adding, for each level $k \in \{0, \ldots, K-3\}$, the edge of the saturator connecting the first vertex of $\sigma|_k$ with the last vertex of $\sigma|_{k+3}$ as shown in the example of Fig. 13. Finally, since the first and the last vertex of σ are on the same face, they can be connected with a curve contained into such a face without introducing intersections.

It is easy to implement the above algorithm to work in linear time by building the lists of vertices for each level while visiting σ. Notice that K is bounded by the number of the vertices of the cycle.

Hence, we can state the following result.

Theorem 2 *Given an n-vertex c-planar 3-cluster cycle σ, there exists an algorithm that computes a c-planar drawing of σ in $O(n)$ time.*

From the above construction we also have the following.

Theorem 3 *A c-planar 3-cluster cycle admits a planar saturator that is the collection of three disjoint paths.*

4 Clusters and Grammars

In this section we characterize the c-planar 3-cluster cycles in terms of formal grammars. Namely, we show that the sequences representing such cycles are those generated by a context-free grammar.

We denote by $\mathcal{L}$ the language of all strings on the alphabet $\{a, b, c\}$ such that each string:

(1) contains at least one instance of each label,

(2) does not contain repeated consecutive letters, and

(3) does not start and end with the same letter.

Observe that $\mathcal{L}$ describes all possible 3-cluster cycles. The following lemma holds:

Lemma 5 *$\mathcal{L}$ is a regular set.*

Proof: The statement can be easily proved by showing that language $\mathcal{L}_1$ ($\mathcal{L}_2$, $\mathcal{L}_3$, respectively) of all the strings on the alphabet $\{a, b, c\}$, such that property 1 (2, 3, respectively) holds, is regular, and the intersection between regular languages is a regular language. In turn, the fact that $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ are regular languages can be easily proved by showing that they admit a regular expression. □

Theorem 4 *The following context-free grammar generates all and only the c-planar 3-cluster cycles:*

$$S \rightarrow Z_0 | Z_3$$

$$Z_0 \rightarrow ABCB | ACBC | BCAC | BACA | CABA | CBAB$$

$$Z_3 \rightarrow ABC | BCA | CAB$$

$$A \rightarrow ABA | ACA | a$$

$$B \rightarrow BAB | BCB | b$$

$$C \rightarrow CAC | CBC | c$$

Proof: The proof exploits the same considerations used to prove Theorem 1. Note that symbol Z_0 generates all 3-cluster cycles σ with $Balance(\sigma) = 0$ and symbol Z_3 generates all 3-cluster cycles with $Balance(\sigma) = 3$. □

Theorem 5 *The language of the c-planar 3-cluster cycles is not regular.*

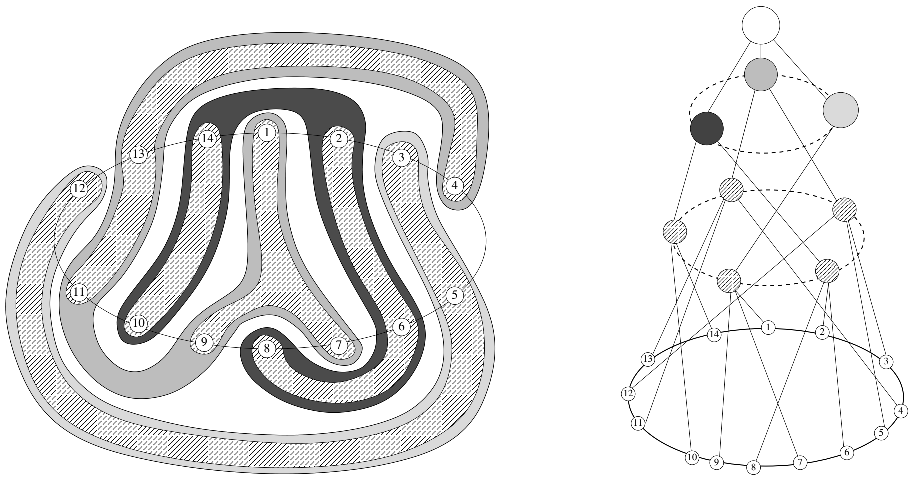
Proof: The proof exploits the equivalence classes of the Myhill-Nerode theorem: given a language L two strings α and β are said to be equivalent if for each string γ, $\alpha\gamma$ and $\beta\gamma$ both belong or both do not belong to L. Language L is regular iff the number of equivalence classes induced by the above equivalence relation is finite [16]. For each integer $n \geq 1$, denote by α_n the string $(abc)^n$. For each pair n, m, with $n < m$, α_n concatenated with $\gamma_n = (acb)^{n-1}$ yields a string of the language (corresponding to a c-planar cycle with balance 3) while α_m concatenated with γ_n yields a string not belonging to the language (corresponding to a c-planar cycle with balance greater than 3). Thus, for $n < m$, α_n and α_m belong to two different equivalence classes. It follows that there is at least one equivalence class $[\alpha_n]$ for each $n \geq 1$ and thus the language of the c-planar 3-cluster cycles is not regular. □

From the above two theorems descends that the language of the c-planar 3-cluster cycles is strictly context-free.

5 Cycles in Cycles of Clusters

In this section we present a generalization of the results of Section 3. First, we generalize the results on 3-cluster cycles to k-cluster cycles. Second, we tackle the general problem of testing the c-planarity of a cycle that is clustered into a cycle of clusters that is in turn clustered into another cycle of clusters, and so on. An example is shown in Fig. 14. Fig. 14.a shows a c-planar clustered graph whose underlying graph is a cycle for which two levels of clusters are defined. Fig. 14.b puts in evidence the inclusion relationships between clusters of a given level and clusters of the level directly above it. The same figure shows also that the clusters of each level form a cycle (dashed edges).

We start by introducing preliminary assumptions and definitions. We consider clustered graphs $C(G, T)$ in which all the leaves of the inclusion tree T have the same distance from the root (we call *depth* such a distance). A clustered graph which has not this property can be easily reduced to this case by inserting "dummy" nodes in T. Hence, from



(a) (b)

Figure 14: A clustered graph where at each level of the inclusion tree the nodes form a cycle. (a) A c-planar drawing. (b) The inclusion tree augmented with dashed edges that put in evidence the adjacencies between nodes at the same level.

now on we consider only inclusion trees whose leaves are all at the same depth. We define as $G^l(V^l, E^l)$ the graph whose vertices are the nodes of T at distance l from its root, and an edge (μ, ν) exists if and only if an edge of G exists incident to both μ and ν.

For example, G^0 has only one vertex and G^L, where L is the depth of the tree, is the underlying graph G of $C(G,T)$. We label each vertex ν of G^l with the cluster (corresponding to a vertex of G^{l-1}) which ν belongs to. If G^l is a cycle, then it is possible to describe G^l with the cyclic sequence of the labels of its vertices. If also G^{l-1} is a cycle, we consider the labels of G^l cyclically ordered according to the order they appear in G^{l-1}. Further, $Balance(G^l)$ can be defined as in Section 3 with values in $\{0, k, 2k, 3k, \ldots\}$, where k is the length of G^{l-1}.

According to the above definitions a 3-cluster cycle is a clustered graph where T as depth 2, G^2 is a cycle and G^1 is a cycle of length 3. In fact, the results of Section 3 can be extended to the case in which G^1 is a cycle of an arbitrary length.

Theorem 6 *Given an n-vertex clustered graph $C(G,T)$, such that T has depth 2 and G^1 and G^2 are cycles, then:*

1. *there exists an algorithm to test if C is c-planar in $O(n)$ time;*
2. *if C is c-planar, a c-planar drawing of C can be computed in $O(n)$ time.*

Proof: The proof exploits the same considerations and constructions of Theorems 1 and 2. If the length of G^1 is k then C is c-planar iff $Balance(G^2) \in \{0, k\}$. In order to find a c-planar drawing of C, if it exists, the same strategy described in Section 3 can be applied, where, since in the construction depicted in Fig. 13 vertices belonging to level j and level $j+k$ belong to the same cluster, an edge of the saturator is added between the first vertex of level j and the last vertex of level $j+k$. □

Let $C(G,T)$ be a clustered graph and l be an integer between 1 and L, where L is the depth of T. A new clustered graph $C^l(G, T^l)$ can be obtained from C by replacing T

with a tree T^l obtained from T by connecting all the nodes at depth l with the root and deleting all the nodes having depth greater than zero and less than l. According to this definition, $C^1 = C$. The c-planarity of C^l can be used to study the c-planarity of C^{l-1}, as is shown in the following lemma.

Lemma 6 *Let $C(G,T)$ be a clustered graph and l be an integer between 1 and L, where L is the depth of T. Let C^l be c-planar, G^l be a cycle, and G^{l-1} be a cycle of length k. C^{l-1} is c-planar iff $Balance(G^l) \in \{0, k\}$.*

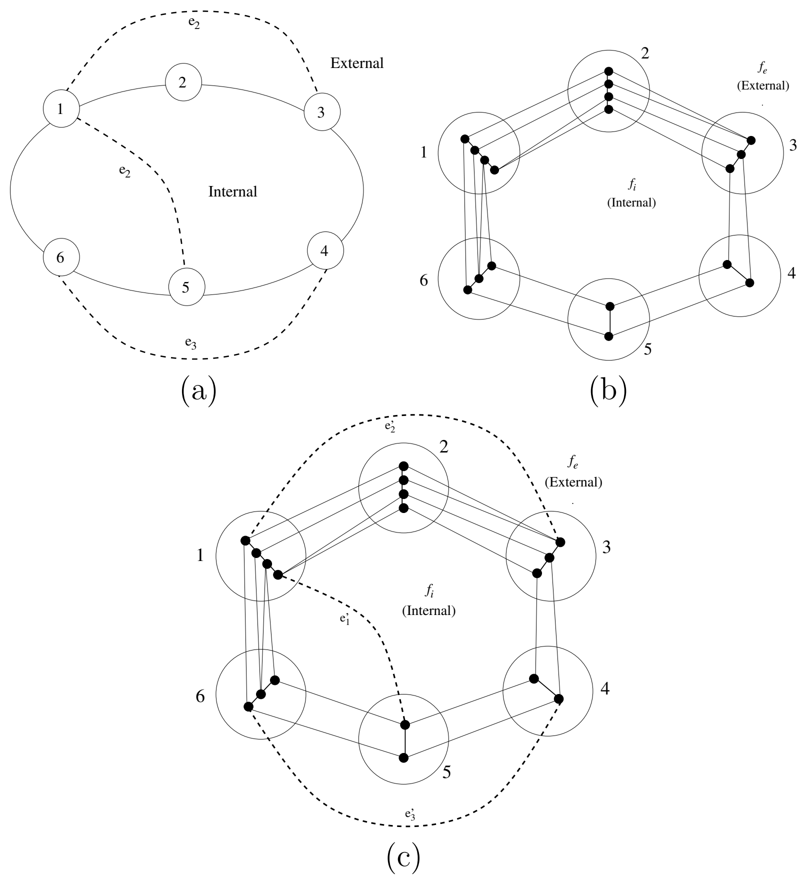


Figure 15: (a) Drawing Γ_{G^l} of G^l with edges of the planar saturator added to the external or internal face. (b) Drawing Γ_{C^l} in which two faces (called the *internal* and *external* face) touching all the clusters can be found. (c) saturator edges joining suitable vertices of Γ_{C^l}.

Proof: First, we prove that if $Balance(G^l) \in \{0, k\}$, then C^{l-1} is c-planar by producing a planar drawing $\Gamma_{C^{l-1}}$ of it. Since $Balance(G^l) \in \{0, k\}$, then it exists a planar drawing Γ_{G^l} of G^l augmented with the edges of a planar saturator connecting vertices of G^l with the same label. The edges of this planar saturator are drawn in Γ_{G^l} either internally or externally with respect to the cycle G^l (see Fig. 15.a).

Let Γ_{C^l} be a planar drawing of the underlying graph G augmented with the edges of a planar saturator of C^l. Such a drawing exists because C^l is c-planar. Two faces of Γ_{C^l} are incident to at least one vertex belonging to ν for each vertex ν of G^l (see Fig. 15.b).We call such faces f_i and f_e, where f_e is the unbounded one. Also, we denote with $v_{f_i,\nu}$ ($v_{f_e,\nu}$) an arbitrary vertex of ν incident to f_i (f_e).

A saturator of C^{l-1} can be constructed from the planar saturator of C^l by adding one edge e' for each edge e of the planar saturator of G^l (see Figure 15.c). Edge e' is added within f_i (f_e) if e is drawn internally (externally) in Γ_{G^l}. Let ν and μ be the end vertices of e. Suppose, without loss of generality, that e is added externally. Edge e' is attached to $v_{f_e,\nu}$ and $v_{f_e,\mu}$. The obtained saturator is planar since the starting drawing Γ_{C^l} is planar, and two edges of the saturator can not intersect in in $\Gamma_{C^{l-1}}$ since they don't intersect in in Γ_{G^l}.

The second part of the proof shows that if C^{l-1} is c-planar then $Balance(G^l) \in \{0, k\}$. Assume that there exists a planar saturator S of C^{l-1}. Consider graph G' obtained by adding to G the edges of the planar saturator S. For each cluster μ that is a vertex of G^l we contract all edges of G' which have both extremes in μ (we use label μ to denote the resulting vertex) and remove multiple edges. Note that since the edges of S make each cluster connected we obtain a new graph G'' with the following properties:

- there is a one-to-one correspondance among the vertices of G^l and the vertices of G'',
- for each edge of G^l there is a correspondent edge in G''.

Since the contraction operation preserves planarity and connectivity, the edges of S that was incident on distinct clusters are still present in G'', such edges connect all vertices with the same label, and G'' is planar. Hence, the edges of G'' which have no corresponding edge in G^l form a planar saturator for G^l. Since G^l admits a planar saturator, its balance is in $\{0, k\}$. □

Lemma 7 *Let $C = (G, T)$ be a clustered graph and let be l an integer between 1 and L, where L is the depth of T. If C^l is not c-planar, then $C^0 = C$ is not c-planar.*

Proof: If C^l is not c-planar, any saturator introduces a subdivision of $K_{3,3}$ or K_5 in the graph G. Since any saturator of C^0 contains a (non planar) saturator of C^l, it is always possible to find an obstruction in the graph G augmented with the edges of the saturator of C^0. Hence C^0 can not be c-planar. □

The following theorems state the main results about c-planarity testing for cluster graphs in which each G^l, $l \in \{0, \ldots, L\}$, is a cycle.

Theorem 7 *Given an n-vertex clustered graph $C(G, T)$, such that T has depth L and, for $l > 0$, G^l is a cycle, there exists an algorithm to test if C is c-planar in $O(Ln)$ time.*

Proof: We apply Lemma 6 to the clustered graphs C^l for $l = L, L-1, \ldots, 2$. Since each test can be performed in $O(n)$ time, the statement follows. □

Theorem 8 *Given an n-vertex clustered graph $C(G, T)$, such that T has depth L and, for $l > 0$, G^l is a cycle, if C is c-planar there exists an algorithm to compute a c-planar drawing of C in $O(Ln)$ time.*

Proof: Since Lemma 6 is proved by construction, by applying, level by level, Lemma 6 starting from level L to level 1, a planar saturator of C can be obtained. Since each step may be performed in $O(n)$ time, the statement follows. □

6 Conclusions and Open Problems

In this paper we studied a peculiar family of non-connected clustered graphs. Namely, we studied clustered graphs whose underlying graph is a simple cycle.

Besides the general problem of stating the complexity of the c-planarity testing of non-connected clustered graphs, several other problems remain open:

- Are there other families of non-connected clustered graphs whose c-planarity can be efficiently assessed and whose underlying graph has a simple structure? For example, what happens if the underlying graph is a tree? It is easy to show that a flat clustered graph whose underlying graph G^2 is a path and such that graph G^1 is a cycle, is c-planar. It is also easy to find an example of a non-connected flat clustered graph whose underlying graph G^2 is a tree, such that G^1 is a cycle and that is not c-planar (see Fig. 16).

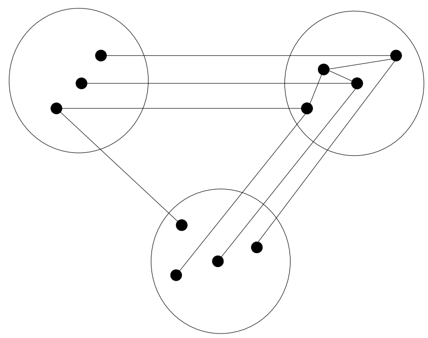


Figure 16: A 3-cluster graph that is not c-planar. The underlying graph is a tree.

- Suppose that the underlying graph has a fixed embedding. Can this hypothesis simplify the c-planarity testing?
- Can the techniques introduced in this paper be combined with techniques known in the literature for devising tools able to handle the c-planarity testing and embedding problem for more complex families of clustered graphs?

References

[1] T. C. Biedl. Drawing planar partitions III: Two constrained embedding problems. Tech. Report RRR 13-98, RUTCOR Rutgen University, 1998.

[2] T. C. Biedl, M. Kaufmann, and P. Mutzel. Drawing planar partitions II: HH-Drawings. In *Workshop on Graph-Theoretic Concepts in Computer Science (WG'98)*, volume 1517, pages 124–136. Springer-Verlag, 1998.

[3] S. Cornelsen and D. Wagner. Completely connected clustered graphs. In *Proc. 29th Intl. Workshop on Graph-Theoretic Concepts in Computer Science (WG 2003)*, volume 2880 of *LNCS*, pages 168–179. Springer-Verlag, 2003.

[4] G. Di Battista, W. Didimo, and A. Marcandalli. Planarization of clustered graphs. In *Proc. Graph Drawing 2001 (GD'01)*, LNCS, pages 60–74. Springer-Verlag, 2001.

[5] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing.* Prentice Hall, Upper Saddle River, NJ, 1999.

[6] G. Di Battista, W. P. Liu, and I. Rival. Bipartite graphs, upward drawings, and planarity. *Information Processign Letters*, 36(6):317–322, 1990.

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

[8] P. Eades and Q. W. Feng. Multilevel visualization of clustered graphs. In Stephen C. North, editor, *Proc. Graph Drawing 1996 (GD'96)*, volume 1190 of *LNCS*, pages 101–112. Springer-Verlag, 1997.

[9] P. Eades, Q. W. Feng, and X. Lin. Straight line drawing algorithms for hierarchical graphs and clustered graphs. In Stephen C. North, editor, *Proc. Graph Drawing 1996 (GD'96)*, volume 1190 of *LNCS*, pages 113–128. Springer-Verlag, 1997.

[10] P. Eades, Q.-W Feng, and H. Nagamochi. Drawing clustered graphs on an orthogonal grid. *J. Graph Algorithms Appl.*, 3(4):3–29, 1999.

[11] S. Even. *Graph Algorithms.* Computer Science Press, Potomac, Maryland, 1979.

[12] Q. W. Feng, R. F. Cohen, and P. Eades. How to draw a planar clustered graph. In Ding-Zhu Du and Ming Li, editors, *Proc. COCOON'95*, volume 959 of *LNCS*, pages 21–30. Springer-Verlag, 1995.

[13] Q. W. Feng, R. F. Cohen, and P. Eades. Planarity for clustered graphs. In P. Spirakis, editor, *Symposium on Algorithms (Proc. ESA '95)*, volume 979 of *LNCS*, pages 213–226. Springer-Verlag, 1995.

[14] M. Freire and P. Rodriguez. A graph-based interface to complex hypermedia structure visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 163–166. ACM Press, 2004.

[15] C. Gutwenger, M. Jünger, S. Leipert, P. Mutzel, M. Percan, and René Weiskircher. Advances in C-planarity testing of clustered graphs. In Stephen G. Kobourov and Michael T. Goodrich, editors, *Proc. Graph Drawing 2002 (GD'02)*, volume 2528 of *LNCS*, pages 220–235. Springer-Verlag, 2002.

[16] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.