Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 13, no. 3, pp. 349–378 (2009)

Efficient C-Planarity Testing for Embedded Flat Clustered Graphs with Small Faces

Giuseppe Di Battista[1] *Fabrizio Frati*[1]

[1]Dipartimento di Informatica e Automazione
Università Roma Tre, Italy

Abstract

Let C be a clustered graph and suppose that the planar embedding of its underlying graph is fixed. Is testing the c-planarity of C easier than in the variable embedding setting? In this paper we give a first contribution towards answering the above question. Namely, we characterize c-planar embedded flat clustered graphs with at most five vertices per face and give an efficient testing algorithm for such graphs. The results are based on a more general methodology that sheds new light on the c-planarity testing problem.

Submitted:	Reviewed:	Revised:	Accepted:
December 2007	May 2008	August 2008	November 2008
	Final:	Published:	
	December 2008	November 2009	

Article type:	Communicated by:
Regular paper	S.-H. Hong and T. Nishizeki

Work partially supported by MUR under Project MAINSTREAM Algorithms for Massive Information Structures and Data Streams.

E-mail addresses: gdb@dia.uniroma3.it (Giuseppe Di Battista) frati@dia.uniroma3.it (Fabrizio Frati)

1 Introduction

Determining the computational complexity of the *c*-planarity testing for clustered graphs is one of the main Graph Drawing challenges. However, despite all the research efforts spent, only for restricted families of clustered graphs polynomial-time testing algorithms have been found, and the general problem is still open.

A brief survey on the problem of testing the *c*-planarity of clustered graphs can be found in [3]. The classes of clustered graphs for which the problem is known to be polynomial-time solvable are the following:

- *c-connected clustered graphs*, in which each cluster induces a connected subgraph of the underlying graph; the first polynomial-time algorithm for this class has been presented in [8].
- *completely connected clustered graphs*, that are *c*-connected clustered graphs such that the complement of the subgraph induced by each cluster is connected; an elegant characterization for this class is shown in [2].
- *almost connected clustered graphs*, in which either all nodes corresponding to non-connected clusters are on the same path in the cluster hierarchy, or for each non-connected cluster its parent and all its siblings are connected [11].
- *extrovert clustered graphs*, a generalization of *c*-connected clustered graphs with special restrictions on the cluster hierarchy [10].
- *cycles of clusters*, in which the hierarchy is *flat*, the underlying graph is a simple cycle, and the clusters are arranged in a cycle [5]; the clustering hierarchy is *flat* if all clusters, but for the root, are at the same level.
- *clustered cycles*, that are clustered graphs in which the hierarchy is flat, the underlying graph is a simple cycle, and the clusters are arranged into an embedded plane graph [4].

Let C be a clustered graph. Suppose that the planar embedding of its underlying graph is fixed. Is testing the *c*-planarity of C easier than in the variable embedding setting? This question is motivated by the existence of many Graph Drawing problems on planar graphs that are in general NP-hard and that become polynomial-time solvable if the embedding is fixed. Testing if a graph admits an orthogonal planar drawing with at most k bends [15, 9] or if a graph admits an upward planar drawing [1, 9] are examples of such problems.

In this paper we give a first contribution towards answering the above question. Namely, we characterize *c*-planar embedded flat clustered graphs with at most five vertices per face and give an efficient testing algorithm for such graphs.

Our approach is to look for an augmentation that adds to the embedded underlying graph extra edges such that the resulting clustered graph is *c*-connected and *c*-planar. We call *candidate saturating edges* those edges that are potential candidates for the augmentation. Two of such edges have a *conflict* if using both of them in the augmentation causes a crossing. We present a characterization for *single-conflict embedded flat clustered graphs*, that are embedded clustered graphs such that (i) the cluster hierarchy is flat and (ii) each candidate saturating edge has a conflict with at most one other candidate saturating edge. The characterization and the algorithm for embedded flat clustered graphs with at most five vertices per face are a consequence of such a more general result.

Observe that a slightly weaker result, namely a quadratic time algorithm for c-planarity on 3-connected graphs with faces of size at most four, was independently discovered by Jelinkova et al. in [12].

The rest of the paper is organized as follows: In Section 2 we give preliminaries. In Section 3 we characterize c-planar single-conflict embedded flat clustered graphs and c-planar embedded flat clustered graphs with at most five vertices per face. In Section 4 we present a linear time and space c-planarity testing algorithm. Section 5 contains conclusions and open problems. A preliminary version of this paper appeared in [7].

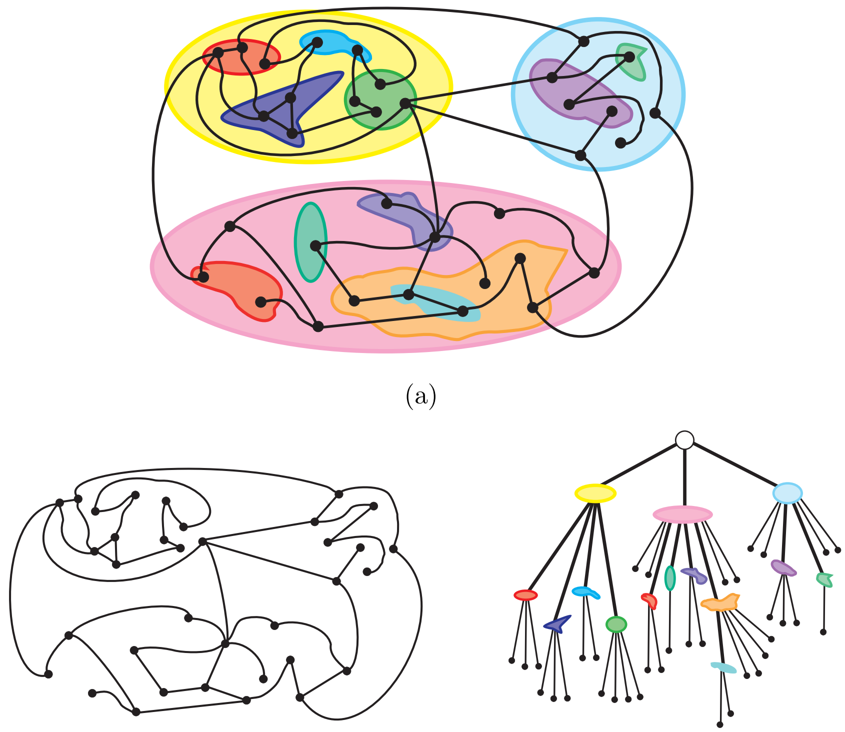
2 Preliminaries

A graph G is *vertex k-connected* (resp. *edge k-connected*) if the removal of any $k-1$ vertices (resp. edges) leaves G connected. A *separating edge* (sometimes also called *bridge*) is an edge whose removal disconnects G.

A *drawing* of a graph is a mapping of each vertex to a distinct point of the plane and of each edge to a Jordan curve between the endpoints of the edge. A *planar drawing* is such that no two edges intersect except, possibly, at common endpoints. A planar drawing of a graph determines a circular ordering of the edges incident to each vertex. Two drawings of the same graph are *equivalent* if they determine the same circular orderings around each vertex. A *planar embedding* (or *combinatorial embedding*) is an equivalence class of planar drawings. A planar drawing partitions the plane into topologically connected regions, called *faces*. The unbounded face is the *outer face*. Two planar drawings with the same combinatorial embedding have the same faces. However, such drawings could still differ for their outer face. The *dual graph D* of a planar embedded graph G is the graph with a vertex for each face of G and with an edge $e(D)$ between two vertices if the corresponding faces share an edge $e(G)$; edge $e(D)$ is *dual* to edge $e(G)$.

In the following we will deal both with biconnected (that is vertex 2-connected) and with simply connected (that is vertex 1-connected) embedded planar graphs. In the former case, the "*number of vertices in a face*" is trivially defined as the number of vertices incident to the face, while in the latter one is meant to be the number of occurrences of vertices on the border of the face.

A *clustered graph* (see Fig. 1.a) is a pair $C(G,T)$, where G is a graph (see Fig. 1.b) and T is a rooted tree (see Fig. 1.c) such that the leaves of T are the vertices of G. Graph G and tree T are called *underlying graph* and *inclusion tree*, respectively. Each internal node μ of T corresponds to the subset $V(\mu)$



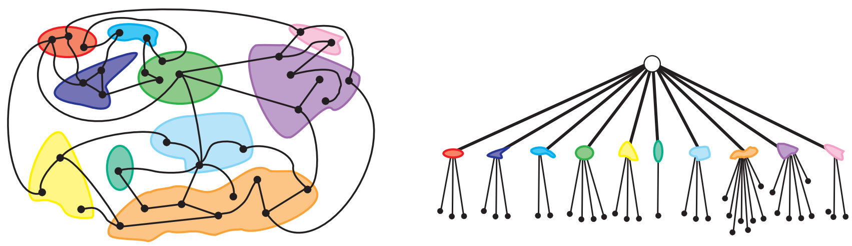
(a)

(b) (c)

Figure 1: (a) A clustered graph $C(G,T)$. (b) The underlying graph G of C. (c) The inclusion tree T of C.

(called *cluster*) of the vertices of G that are leaves of the subtree of T rooted at μ; the subgraph of G induced by the vertices in $V(\mu)$ is denoted by $G(\mu)$. If each cluster induces a connected subgraph of G, then C is *c-connected*, otherwise C is non-c-connected. In the latter case each cluster generally induces several connected components of G. The clustered graph in Fig. 1.a is non-c-connected. An *embedded clustered graph* is a clustered graph such that G is connected and embedded, that is, the combinatorial embedding of the underlying graph of C is fixed.

A drawing of a clustered graph $C(G,T)$ consists of a drawing of G and of a representation of each node μ of T as a simple closed region $R(\mu)$ such that: (i) $R(\mu)$ contains the drawing of $G(\mu)$; (ii) $R(\mu)$ contains a region $R(\nu)$ iff ν is a descendant of μ in T; and (iii) the borders of any two regions do not intersect. Consider an edge e and a node μ of T. If e crosses the boundary of $R(\mu)$ more than once, we say that edge e and region $R(\mu)$ have an *edge-region crossing*. A drawing of a clustered graph is *c-planar* if it does not have edge crossings or edge-region crossings. The drawing in Fig. 1.a is c-planar. A clustered graph is c-planar if it admits a c-planar drawing. An embedded clustered graph is c-planar if it admits a c-planar drawing in which the embedding of G is preserved.



(a) (b)

Figure 2: (a) A flat clustered graph $C(G, T)$. (b) The inclusion tree T of C has height three.

A *flat clustered graph* is a clustered graph such that in any path from the root to a leaf of T there are at most three nodes. The clustered graph in Fig. 2 is flat, while the one in Fig. 1 is not. To simplify the notation, when referring to a flat clustered graph, given a vertex v of the underlying graph we say that the *cluster of* v is its parent in T. Also, we call *clusters* only the children of the root.

Consider an embedded flat clustered graph $C(G, T)$. For each face f of G a set of *candidate saturating edges* is defined as follows: Let O be the clockwise circular order of the vertices on the border of f. Subdivide such vertices into subsets such that each subset V_i contains a maximal sequence of consecutive vertices in O belonging to the same cluster. Introduce a candidate saturating edge for each two subsets $V_i \neq V_j$ such that (i) V_i and V_j contain vertices of the same cluster μ_k and (ii) V_i and V_j are in different connected components of $G(\mu_k)$. Candidate saturating edges represent edges that can be added to the embedded clustered graph to make the subgraph induced by each cluster connected (see Fig. 3.a and 3.b).

For a cluster μ_i of T we define $\mathcal{G}_i$ as the embedded multigraph whose vertices are the connected components of $G(\mu_i)$ and whose edges are the candidate saturating edges connecting such components. The embedding of $\mathcal{G}_i$ is given by the order of the faces around the vertices of G (Fig. 3.c, 3.d, and 3.e). Observe that $\mathcal{G}_i$ does not have self-loops and is, in general, non-planar. However, possible crossings are only between edges introduced in the same face of G.

Two candidate saturating edges e_1, joining connected components $G_1(\mu_i)$ and $G_2(\mu_i)$ of $G(\mu_i)$, and e_2, joining connected components $G_1(\mu_j)$ and $G_2(\mu_j)$ of $G(\mu_j)$, with $\mu_i \neq \mu_j$ and with e_1 and e_2 in the same face f of G, have a *conflict* if $G_1(\mu_i)$, $G_1(\mu_j)$, $G_2(\mu_i)$, and $G_2(\mu_j)$ appear in this order around the border of f. Informally speaking, two candidate saturating edges have a conflict if adding both of them to the clustered graph causes a crossing.

The following theorem shows the role of the candidate saturating edges of a flat embedded clustered graph C in the c-planarity of C. Even if not explicitly

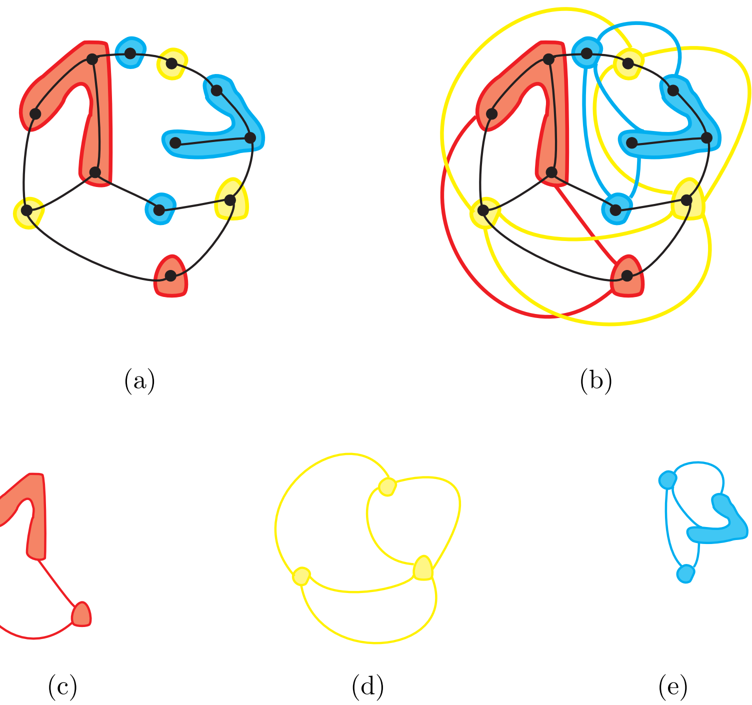


Figure 3: (a) An embedded flat clustered graph C. Different clusters have different colors. The connected components of each cluster are inside simple connected regions having the color of the cluster. (b) Clustered graph C and its candidate saturating edges. Candidate saturating edges of each cluster have the same color of the cluster. (c)–(d)–(e) Multigraphs $\mathcal{G}_i$ for C. The vertices of $\mathcal{G}_i$ are the connected components of $G(\mu_i)$ and the edges of $\mathcal{G}_i$ are the candidate saturating edges connecting such components.

stated, Theorem 1 has been already used in [5].

Theorem 1 *An embedded flat clustered graph $C(G,T)$ is c-planar if and only if: (1) G is planar; (2) there exists a face f in G such that when f is chosen as outer face for G no cycle composed by vertices of the same cluster encloses a vertex of a different cluster; (3) it is possible to augment G to a graph G' by adding a subset of the candidate saturating edges of C so that no two added edges have a conflict and so that clustered graph $C'(G',T)$ is c-connected.*

Proof: First, we prove the necessity. The necessity of Condition 1 is trivial.

The necessity of Condition 2 easily descends from the definition of c-planarity. Namely, suppose that any plane embedding of G contains a cycle $\mathcal{C}$ composed by vertices belonging to cluster μ_i, such that $\mathcal{C}$ encloses a vertex v not belonging to μ_i. By definition of c-planar drawing, the region $R(\mu_i)$ representing μ_i in any drawing $\Gamma(C)$ of C contains $\mathcal{C}$, and hence either $R(\mu_i)$ is not simple, or it contains v, that does not belong to μ_i. By definition of c-planar drawing, in both cases $\Gamma(C)$ is not c-planar.

To prove that Condition 3 is necessary for the c-planarity of C, consider any c-planar drawing $\Gamma(C)$ of C. We show that it is possible to draw candidate saturating edges augmenting G to a graph G' so that the subgraph induced by each cluster in G' is connected and so that the augmented drawing $\Gamma'(C)$ is still c-planar.

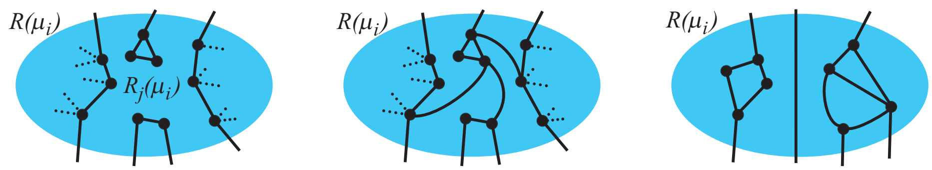


Figure 4: Illustrations for the proof of Theorem 1

Consider the region $R(\mu_i)$ representing in $\Gamma(C)$ a cluster μ_i. Subdivide $R(\mu_i)$ into connected open regions $R_j(\mu_i)$ delimited by the border of $R(\mu_i)$ and by the edges of G. Consider any region $R_j(\mu_i)$ that has on its border vertices of more than one connected component of $G(\mu_i)$. Edges connecting vertices of different connected components can be drawn inside $R_j(\mu_i)$ so that the planarity of the drawing of G is maintained and so that the connected components of $G(\mu_i)$ appearing on the border of $R_j(\mu_i)$ form a unique connected component (see Figs. 4.a and 4.b). Notice that added edges are candidate saturating edges of C. After this step is repeated for every $R_j(\mu_i)$ all the connected components of $G(\mu_i)$ form a unique connected component. In fact, having two connected components in $\Gamma'(C)$ would imply that there is an edge-region crossing in $\Gamma(C)$ (see Fig. 4.c). After the augmentation is performed for every cluster μ_i the set of edges added to G satisfies the properties of Condition 3. Namely, no two added edges have a conflict since edges added to connect $G(\mu_i)$ and $G(\mu_j)$ for different clusters μ_i and μ_j are drawn inside non-overlapping regions $R(\mu_i)$ and $R(\mu_j)$.

Now we prove the sufficiency of Conditions 1, 2, and 3 for the c-planarity of C. Consider any planar drawing Γ of G in which no cycle composed by vertices of the same cluster encloses a vertex of a different cluster (such a drawing exists by Conditions 1 and 2). Consider a set S of candidate saturating edges of C satisfying Condition 3. Insert each edge e of S in Γ inside the face of G for which e is a candidate saturating edge. Since no two edges of S conflict each other, it is possible to do such an insertion so that the resulting drawing Γ' of the augmented graph G' is planar.

As long as G' has at least one edge e^* of S belonging to a cycle in which all vertices are in the same cluster, remove e^* from G' and from Γ'. Clearly, such a removal leaves each cluster connected in G'. Moreover, after all such deletions no edge of any cycle in which all vertices are in the same cluster belongs to S.

For any cluster μ draw a region $R(\mu)$ representing μ in Γ' as a simple closed connected region surrounding the entire drawing of $G'(\mu)$. The border of $R(\mu)$ can be drawn so close to the border of the outer face of $G'(\mu)$ that (i) $R(\mu)$ does not enclose vertices that are outside the outer face of $G'(\mu)$, (ii) the border of $R(\mu)$ does not touch edges that are not incident to vertices of the outer face of $G'(\mu)$, and (iii) the borders of any two clusters do not intersect.

We prove that the resulting clustered drawing $\Gamma(C)$ of C is c-planar. By Condition 1, the drawing of G is planar. By construction, for each cluster μ, region $R(\mu)$ contains the drawing of $G'(\mu)$ in its interior. Suppose that a region $R(\mu)$ encloses a vertex $v \in V(\nu)$, with $\mu \neq \nu$. By the construction of region $R(\mu)$, this implies that there exists a cycle in $G'(\mu)$ enclosing v. However, since every cycle of G' in which all vertices are in the same cluster is also a cycle of G, this would imply that Condition 2 is not satisfied by C. By the construction of regions $R(\mu)$ no two borders of different clusters intersect in $\Gamma(C)$. Finally, an edge-region crossing would imply an edge crossing in G', that is planar by Condition 3 and by the definition of saturator. □

Hence, given an embedded flat clustered graph $C(G, T)$, if Conditions 1 and 2 are satisfied by G, the problem of testing the c-planarity of C can be restated as the problem of testing if it is possible to select from multigraphs $\mathcal{G}_i$ a set of candidate saturating edges to enforce Condition 3 of Theorem 1. If such a set exists, we call it a *saturator* of C.

Lemma 1 *An embedded flat clustered graph $C(G, T)$ admits a saturator if and only if it admits an acyclic saturator.*

Proof: Consider any saturator S of C and denote by G' the embedded graph obtained by inserting each edge e of S inside the face of G for which e is a candidate saturating edge. As long as G' has at least one edge e^* of S belonging to a cycle in which all vertices are in the same cluster, remove e^* from G'. After the removal the edges added to G are still a saturator of C, since, for each cluster μ, $G'(\mu)$ is connected and since the c-planarity of $C(G' \setminus e^*, T)$ is a consequence of the c-planarity of $C(G', T)$. Finally, observe that after all such deletions are performed no cycle composed of edges all belonging to S exists in G'. □

Hence, the problem of testing if an embedded flat clustered graph satisfying Conditions 1 and 2 of Theorem 1 is c-planar is reduced to the one of testing if there exists a spanning tree of each $\mathcal{G}_i$ such that no two edges in different spanning trees have a conflict.

3 A Characterization

We restrict ourselves to those embedded flat clustered graphs in which each candidate saturating edge has a conflict with at most one other candidate saturating edge. We call an embedded flat clustered graph satisfying such a property to be *single-conflict*. The clustered graph of Fig. 5 is single-conflict, while the one of Fig. 3 is not.

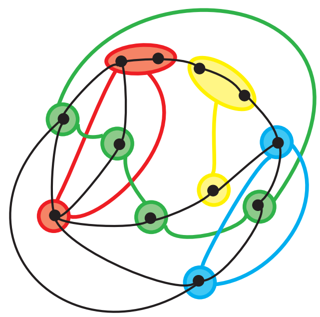
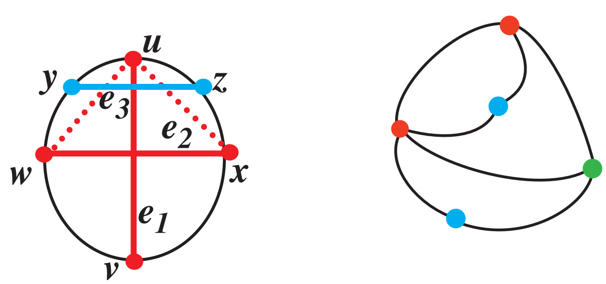


Figure 5: A single-conflict flat embedded clustered graph.

Consider a single-conflict embedded flat clustered graph $C(G,T)$ and, for any cluster μ_i in T, consider multigraph $\mathcal{G}_i$. We have the following structural lemma, showing that if two edges $e_1 = (u,v)$ and $e_2 = (x,w)$ of $\mathcal{G}_i$ cross, that is, vertices u, x, v, and w appear in this order on the border of the face f for which e_1 and e_2 are candidate saturating edges, then none of e_1 and e_2 can possibly cross an edge e_3 of a multigraph $\mathcal{G}_j$, with $i \neq j$.

Lemma 2 *If a graph $\mathcal{G}_i$ contains two crossing edges e_1 and e_2, then e_1 and e_2 have no conflict with edges of other multigraphs.*



(a) (b)

Figure 6: (a) Illustration for the proof of Lemma 2. (b) Illustration for the proof of Lemma 3. Graph $\mathcal{G}_i$ for the cyan cluster is not connected and there is no way of adding edges to the clustered graph to make the cyan cluster connected.

Proof: Suppose, for a contradiction, that (i) C is a single-conflict embedded flat clustered graph, (ii) e_1 and e_2 are edges of $\mathcal{G}_i$, that is, e_1 and e_2 are candidate saturating edges for a cluster μ_i, (iii) e_1 and e_2 cross inside a face f of G, and

(iv) e_1 has a conflict with an edge e_3 of a multigraph $\mathcal{G}_j$, with $j \neq i$, inside f (see Fig 6.a).

We claim that e_3 has conflicts with at least two edges of $\mathcal{G}_i$ and hence C is not a single-conflict embedded clustered graph. Let u and v, w and x, and y and z be the end-vertices of e_1, e_2, and e_3, respectively. If e_3 crosses e_2, the statement follows. Otherwise we can suppose without loss of generality, up to a renaming of the vertices, that w, y, u, z, x, and v appear in this order around f. If vertices u and w do not belong to the same connected component of $G(\mu_i)$, then there exists in $\mathcal{G}_i$ an edge joining u and w that has a conflict with e_3 and the statement follows. Analogously, if vertices u and x do not belong to the same connected component of $G(\mu_i)$, then there exists in $\mathcal{G}_i$ an edge joining u and x that has a conflict with e_3 and the statement follows. However, either u and w, or u and x belong to different connected components of $G(\mu_i)$, otherwise u, w, and x would be in the same connected component of $G(\mu_i)$ and e_2 would not be a candidate saturating edge. □

By Lemma 3, we can assume that in the interesting cases the $\mathcal{G}_i$'s are connected (see Fig. 6.b).

Lemma 3 *If there exists a multigraph $\mathcal{G}_i$ that is not connected, then C is not c-planar.*

Proof: If a multigraph $\mathcal{G}_i$ is not connected, then adding to G any set of candidate saturating edges would not connect $G(\mu_i)$. Hence, by Theorem 1, C is not c-planar. □

There are edges in the $\mathcal{G}_i$'s that must be used in any saturator of C. Conversely, there are edges that will never be used in any saturator. Further, there are edges that can be supposed to belong to a saturator without altering the possibility to have one. Roughly speaking, such edges do not belong to the "core" of the problem. Hence, in the following we simplify the $\mathcal{G}_i$'s with an algorithm that either returns that C is not c-planar or returns a structure where there are no trivial choices. For this purpose, we define two operations on $\mathcal{G}_i$, that remove or collapse edges, to be used in the *simplification phase.*

The operation of *removing* an edge e from $\mathcal{G}_i$, corresponds to the choice of not using e as an edge of the saturator of C. Notice that, when an edge e is removed from $\mathcal{G}_i$, an edge of $\mathcal{G}_j$, with $i \neq j$, that eventually had a conflict with e does not have a conflict any longer.

The operation of *collapsing* an edge e with end-vertices u and v in $\mathcal{G}_i$ corresponds to the choice of using e as an edge of the saturator of C. It consists of (see Fig. 7): (i) deleting vertices u and v, (ii) removing from $\mathcal{G}_i$ all edges between u and v, and (iii) inserting in $\mathcal{G}_i$ a new vertex w whose incident edges are those of u and v. The embedding of $\mathcal{G}_i$ is preserved. The collapsing operation "preserves" the conflicts. Namely, let e_i be an edge of $\mathcal{G}_i$ incident to u or to v but not to both. Suppose that e_i has a conflict (has not a conflict) with an edge e_j of $\mathcal{G}_j$, with $i \neq j$. After collapsing edge e in a new vertex w the edge incident to w corresponding to e_i has a conflict (resp. has not a conflict) with e_j. When an edge e is collapsed, the edge that conflicts with e, if any, is removed. In fact, collapsing e corresponds to choosing it in a saturator, hence no edge conflicting with e can be introduced in the saturator.

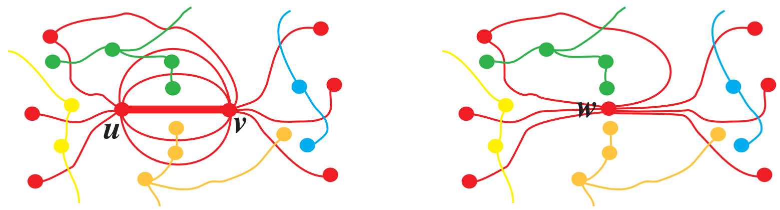


Figure 7: The operation of collapsing an edge (u, v): (a) Before collapsing (u, v). (b) After collapsing (u, v).

The simplification phase is as follows. Repeatedly modify the $\mathcal{G}_i$'s by applying one of the following simplifications. From now on, $\mathcal{G}_i$ denotes the multigraph obtained from the starting $\mathcal{G}_i$ after some simplifications have been performed.

Simplification 1: If there exists an edge e of a multigraph $\mathcal{G}_i$ that has no conflict, then collapse e in $\mathcal{G}_i$.

Simplification 2: If there exist a separating edge e_i and a non-separating edge e_j that are in multigraphs $\mathcal{G}_i$ and $\mathcal{G}_j$, respectively, and that conflict each other, then collapse e_i in $\mathcal{G}_i$ and remove e_j from $\mathcal{G}_j$.

Simplification 3: If there exist two separating edges e_i and e_j that are in multigraphs $\mathcal{G}_i$ and $\mathcal{G}_j$, respectively, and that conflict each other, then stop because C is not c-planar.

If the algorithm does not stop for non-c-planarity, we call the final multigraph $\mathcal{G}_i$ *candidate saturating graph* for cluster μ_i and we denote it by $\mathcal{G}_i^*$. Also, we say that μ_i *admits a candidate saturating graph.*

Observe that the above operations modify graphs $\mathcal{G}_i$. However, at any step of the simplification phase each edge e of $\mathcal{G}_i$ is associated with two vertices u and v and a face f of G meaning that if e is chosen to be in a saturator an edge between u and v is inserted in f. We preprocess $\mathcal{G}_i$ labeling each edge with its endpoints and with a face. Such labels are never changed by the operations described below. In the following, each time we add an edge e of $\mathcal{G}_i$ to a saturator, we add to G an edge between the endpoints and within the face specified by the label of e.

The following properties hold.

Property 1 *None of Simplifications 1, 2, and 3 could disconnect any multigraph $\mathcal{G}_i$.*

Proof: Simplification 1 collapses an edge of a multigraph $\mathcal{G}_i$. If $\mathcal{G}_i$ was connected before such a simplification, then it is still connected after that. Further, no edges of other multigraphs are removed when applying Simplification 1. Simplification 2 collapses an edge e_i of a multigraph $\mathcal{G}_i$ and removes the edge e_j of a multigraph $\mathcal{G}_j$ that had a conflict with e_i. However, if $\mathcal{G}_i$ was connected before such a simplification, then it is still connected after that, and since e_j is not a separating edge, then $\mathcal{G}_j$ remains connected after Simplification 2. Simplification 3 does not modify and hence does not disconnect any multigraph $\mathcal{G}_i$. □

Property 2 *None of Simplifications 1, 2, and 3 can create a self-loop in any multigraph* $\mathcal{G}_i$.

Proof: A self-loop in a multigraph $\mathcal{G}_i$ cannot be created by a removing operation. Further, when an edge e of a multigraph $\mathcal{G}_i$ is collapsed in a vertex w, edges parallel to e are removed. Hence, no self-loop is inserted in $\mathcal{G}_i$. □

Property 3 *The subgraphs induced by the collapsed edges are acyclic.*

Proof: Suppose that the subgraph induced by the set E of collapsed edges contains a cycle $\mathcal{C}$. Consider the last simplification s_m of the simplification phase that collapses one of the edges of $\mathcal{C}$, say edge $e = (u, v)$. A path $\mathcal{P}$ connecting u and v exists in E composed of candidate saturating edges that have been collapsed before s_m. The edges of $\mathcal{P}$ are collapsed in a single vertex w at the beginning of step s_m. By Property 2, vertex w has no self-loops, hence no edge connecting two vertices of $\mathcal{P}$ exists at step s_m. □

Property 4 *Candidate saturating graphs are planar embedded and edge* 2*-connected.*

Proof: Each multigraph $\mathcal{G}_i$ before the simplification phase is planar embedded and the operations of removing and collapsing edges of $\mathcal{G}_i$ leave $\mathcal{G}_i$ planar embedded. By Property 1, multigraph $\mathcal{G}_i^*$ is connected. Further, if it has a separating edge e, then either e would be chosen to be in a saturator by one of Simplifications 1 and 2 (depending on whether e has no conflict or has a conflict with a non-separating edge) or C would not admit candidate saturating graphs (if e has a conflict with a separating edge). □

Property 5 *Any edge of a candidate saturating graph has exactly one conflict with an edge of a different candidate saturating graph.*

Proof: Any edge of a candidate saturating graph has at most one conflict with an edge of a different candidate saturating graph, since the embedded flat clustered graph is assumed to be single-conflict and operations of removing and collapsing edges do not introduce new conflicts. Any edge of a candidate saturating graph has at least one conflict with an edge of a different candidate saturating graph, otherwise it would be chosen to be in a saturator by Simplification 1. □

We now give lemmas in order to prove that each simplification performed by the algorithm preserves the possibility of finding a saturator of C. Consider simplification s_m, that is performed at a certain step of the simplification phase. Simplification s_m can be one of Simplifications 1, 2, or 3. Denote by $s_0, s_1, \ldots, s_{m-1}$ the simplifications that have been performed before s_m. Denote also by E the set of edges that have been collapsed while applying $s_0, s_1, \ldots, s_{m-1}$. Inductively, suppose that if an acyclic saturator of C exists, then there exists an acyclic saturator composed only of the edges of E plus some of the edges remaining in the $\mathcal{G}_i$'s after simplifications $s_0, s_1, \ldots, s_{m-1}$. This is indeed the case when no simplification has been performed yet.

Lemma 4 *Consider an edge e of $\mathcal{G}_i$ with no conflict. We have that C admits a saturator only if it admits an acyclic saturator containing e and containing the edges of E.*

Proof: Suppose C admits a saturator. Then, by Lemma 1, it admits an acyclic saturator. Moreover, by inductive hypothesis, it admits an acyclic saturator S such that $E \subseteq S$. If S contains e the statement follows. Otherwise, observe that since S is a saturator, there exists a set $S' \subseteq S$ of edges forming a path between the end-vertices u and v of e. Hence, the edges of $S' \cup \{e\}$ form a cycle. Notice that not all the edges of S' belong to E, otherwise u and v would not have been distinct vertices in $\mathcal{G}_i$ after simplifications $s_0, s_1, \ldots, s_{m-1}$. Hence, the set S^* of edges obtained from S by inserting e and by removing any edge of S' not in E is an acyclic saturator of C containing E and e. Namely, all the connected components of C are connected by a path of edges in S^* and since e has no conflict and S is a saturator, no two edges in S^* have a conflict. □

Lemma 5 *Consider two edges e_i and e_j of two distinct multigraphs $\mathcal{G}_i$ for cluster μ_i and $\mathcal{G}_j$ for cluster μ_j, respectively. Suppose that e_i and e_j conflict each other. Also, suppose that e_i is a separating edge, while e_j is not. Then C admits a saturator only if it admits an acyclic saturator containing e_i, containing E, and not containing e_j.*

Proof: Suppose C admits a saturator. Then, by Lemma 1, it admits an acyclic saturator. Moreover, by inductive hypothesis, it admits an acyclic saturator S such that $E \subseteq S$. Since at step s_m end-vertices u and v of e_i are in $\mathcal{G}_i$, no path composed by edges of E connects u and v. Moreover, since e_i is a separating edge, if e_i is not in S adding the edges of S to G would not connect $G(\mu_i)$. Hence $e_i \in S$. Since no two conflicting edges can be simultaneously in S, $e_j \notin S$. □

Lemma 6 *Consider two separating edges e_i and e_j of two distinct multigraphs $\mathcal{G}_i$ for cluster μ_i and $\mathcal{G}_j$ for cluster μ_j, respectively. Suppose that e_i and e_j conflict each other. We have that C is not c-planar.*

Proof: Suppose, for a contradiction, that C admits a saturator. Then, by inductive hypothesis, it admits an acyclic saturator S such that $E \subseteq S$. Since at step s_m the end-vertices u and v of e_i (the end-vertices w and x of e_j) are in $\mathcal{G}_i$ (are in $\mathcal{G}_j$), no path composed by edges of E connects u and v (connects w and x). Moreover, since e_i and e_j are separating edges, if e_i (e_j) is not in S, adding the edges of S to G would not connect $G(\mu_i)$ ($G(\mu_j)$). However, S cannot contain both e_i and e_j, that conflict each other. □

Let μ_i and μ_j be two distinct clusters admitting candidate saturating graphs $\mathcal{G}_i^*$ and $\mathcal{G}_j^*$, respectively. We define graph $\mathcal{G}_{i,j}^*$ as the planar embedded subgraph of $\mathcal{G}_i^*$ induced by the edges having a conflict with the edges of $\mathcal{G}_j^*$. We have (see Fig. 8):

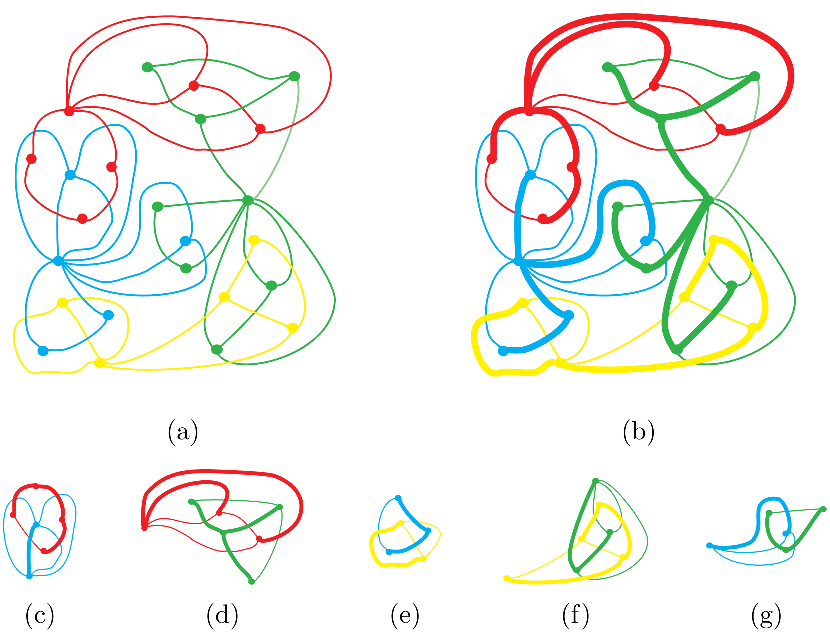


Figure 8: Illustrations for the statement of Theorem 2. (a) A set of candidate saturating graphs $\mathcal{G}_i^*$ for a single-conflict embedded flat clustered graph C. (b) A saturator of C. (c–g) Each picture contains graphs $\mathcal{G}_{i,j}^*$, $\mathcal{G}_{j,i}^*$, and spanning trees $\mathcal{T}_{i,j}^*$, $\mathcal{T}_{j,i}^*$ (in bold).

Theorem 2 *A single-conflict embedded flat clustered graph $C(G,T)$ is c-planar iff:*

1. *G is planar;*
2. *There exists a face f in G such that when f is chosen as outer face for G no cycle composed by vertices of the same cluster encloses a vertex of a different cluster;*
3. *Each cluster of C admits a candidate saturating graph;*
4. *For each pair of distinct clusters μ_i and μ_j, $\mathcal{G}_{i,j}^*$ is edge 2-connected; and*

5. *For each pair of distinct clusters μ_i and μ_j, $\mathcal{G}^*_{i,j}$ is dual to $\mathcal{G}^*_{j,i}$.*

Proof: First, we remark that each vertex of $\mathcal{G}^*_i$ corresponds to a distinct connected component of $G(\mu_i)$ after the edges chosen during the simplification phase have been added into the corresponding faces of G and that an edge connecting vertices u and v of $\mathcal{G}^*_i$ corresponds to an edge connecting a vertex of the connected component corresponding to u to a vertex of the connected component corresponding to v inside a face of G. Since the simplification phase preserves the possibility of finding an acyclic saturator S, then S can be found only if a set of edges can be selected from graphs $\mathcal{G}^*_i$ so that, after the edges of S are inserted into the faces of G, all clusters induce connected graphs, no cycle composed of vertices of the same cluster has been created, and no two edges intersect. It follows that, in order to obtain an acyclic saturator S of C, a spanning tree of each $\mathcal{G}^*_i$ has to be selected such that no two edges of spanning trees of different graphs $\mathcal{G}^*_i$ and $\mathcal{G}^*_j$ have a conflict.

Let S be any acyclic saturator of C and let u and v be any two distinct vertices of any candidate saturating graph $\mathcal{G}^*_i$. We denote by $S(u,v)$ the unique path connecting u and v in the spanning tree of $\mathcal{G}^*_i$ contained in S. We remark that such a path exists, otherwise cluster μ_i would not induce a connected graph after adding the edges of S to G, and is unique, otherwise the chosen saturator S would not be acyclic. If edges e_i and e_j of different candidate saturating graphs $\mathcal{G}^*_i$ and $\mathcal{G}^*_j$ conflict each other, we write $e_i \oplus e_j$.

The necessity of Conditions 1 and 2 descends from the necessity of Conditions 1 and 2 of Theorem 1. We prove the necessity of Condition 3. Suppose that C does not admit candidate saturating graphs. Two cases are possible: Either before the simplification phase one of the $\mathcal{G}_i$'s is not connected, or during the simplification phase two separating conflicting edges are found. In the former case the non-c-planarity of C descends from Lemma 3, in the latter case from Lemma 6.

Now we deal with Condition 4. Suppose that $\mathcal{G}^*_{i,j}$ is not connected and denote by u_1 and u_2 vertices in different connected components. Suppose, for a contradiction, that an acyclic saturator S of C exists. Consider path $S(u_1,u_2)$ (see Fig. 10.a). Since u_1 and u_2 are in different connected components of $\mathcal{G}^*_{i,j}$, there exists an edge $(u_3,u_4) \in S(u_1,u_2)$ such that $(u_3,u_4) \oplus (w_1,w_2)$, where $(w_1,w_2) \in \mathcal{G}^*_k$, with $k \neq i,j$. Consider path $S(w_1,w_2)$. Each edge of $S(w_1,w_2)$ cannot have a conflict with any edge of $S(u_1,u_2)$, otherwise S would contain two conflicting edges, and neither can it have a conflict with any edge (v_1,v_2) of $\mathcal{G}^*_{j,i}$, otherwise (v_1,v_2) would conflict with two candidate saturating edges. Hence, $\mathcal{G}^*_{j,i}$ has at least two connected components. Let v_3 and v_4 be two vertices in such components, respectively. Then, $S(v_3,v_4)$ either contains an edge (v_5,v_6) such that $(v_5,v_6) \oplus (w_3,w_4)$, with $(w_3,w_4) \in S(w_1,w_2)$, implying that S contains two conflicting edges, or contains an edge (v_5,v_6) conflicting with edge (w_1,w_2), implying that (w_1,w_2) conflicts with two candidate saturating edges.

Now suppose that $\mathcal{G}^*_{i,j}$ has a separating edge (u_1,u_2). By construction $(u_1,u_2) \oplus (v_1,v_2)$, where $(v_1,v_2) \in \mathcal{G}^*_{j,i}$. Suppose, for a contradiction, that a saturator S of C exists. Fig. 9 shows the strategy of the proof of such a contra-

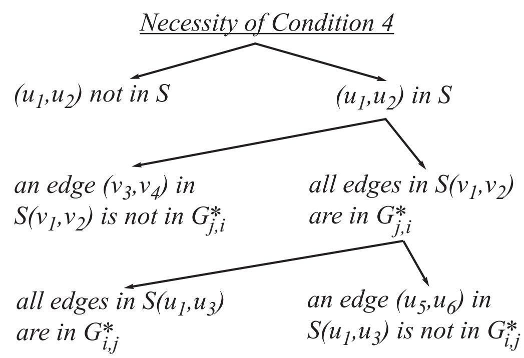


Figure 9: Proof of the necessity of Condition 4. Edge (u_1, u_2) is a separating edge that has a conflict with an edge (v_1, v_2). If $(u_1, u_2) \in S$ and all edges of $S(v_1, v_2)$ belong to $\mathcal{G}^*_{j,i}$, then (u_3, u_4) is an edge that has a conflict with an edge of $S(v_1, v_2)$. Vertices u_1 and u_3 are both internal to cycle $S(v_1, v_2) \cup (v_1, v_2)$.

diction.

- If $(u_1, u_2) \notin S$, then consider $S(u_1, u_2)$ (see Fig. 10.b). Since (u_1, u_2) is a separating edge for $\mathcal{G}^*_{i,j}$, there exists an edge $(u_3, u_4) \in S(u_1, u_2)$ such that $(u_3, u_4) \oplus (w_1, w_2)$, where $(w_1, w_2) \in \mathcal{G}^*_k$, with $k \neq i, j$. Hence, $S(w_1, w_2)$ either contains an edge (w_3, w_4) such that $(w_3, w_4) \oplus (u_5, u_6)$, with $(u_5, u_6) \in S(u_1, u_2)$, implying that S contains two conflicting edges, or contains an edge (w_3, w_4) conflicting with (u_1, u_2), implying that (u_1, u_2) conflicts with two candidate saturating edges.

- If $(u_1, u_2) \in S$, then consider $S(v_1, v_2)$.

  - If an edge $(v_3, v_4) \in S(v_1, v_2)$ is such that $(v_3, v_4) \oplus (w_1, w_2)$, where $(w_1, w_2) \in \mathcal{G}^*_k$, with $k \neq i, j$, a contradiction is obtained as in the previous case (see Fig. 10.c).

  - Otherwise, all edges of $S(v_1, v_2)$ belong to $\mathcal{G}^*_{j,i}$. Consider any edge $(v_3, v_4) \in S(v_1, v_2)$ and edge $(u_3, u_4) \in \mathcal{G}^*_{i,j}$ such that $(u_3, u_4) \oplus (v_3, v_4)$. Let u_1 (u_3) be the endpoint of (u_1, u_2) (resp. of (u_3, u_4)) inside cycle $S(v_1, v_2) \cup (v_1, v_2)$.

    * If $u_1 = u_3$ or if all edges of $S(u_1, u_3)$ have conflicts with edges of $\mathcal{G}^*_{j,i}$ (see Fig. 10.d), consider path $S(u_2, u_4)$. Then there exists an edge $(u_5, u_6) \in S(u_2, u_4)$ such that $(u_5, u_6) \oplus (w_1, w_2)$, where $(w_1, w_2) \in \mathcal{G}^*_k$, with $k \neq i, j$, otherwise (u_1, u_2) would not be a separating edge. Hence, $S(w_1, w_2)$ either contains an edge (w_3, w_4) such that $(w_3, w_4) \oplus (u_7, u_8)$, with $(u_7, u_8) \in S(u_3, u_4)$, implying that S contains two conflicting edges, or an edge (w_3, w_4)

such that $(w_3, w_4) \oplus (u_3, u_4)$ implying that (u_3, u_4) conflicts with two candidate saturating edges.

* If $u_1 \neq u_3$ and $S(u_1, u_3)$ contains at least one edge (u_5, u_6) such that $(u_5, u_6) \oplus (w_1, w_2)$, where $(w_1, w_2) \in \mathcal{G}_k^*$, with $k \neq i, j$ (see Fig. 10.e), then $S(w_1, w_2)$ contains: (i) either an edge (w_3, w_4) such that $(w_3, w_4) \oplus (v_5, v_6)$, with $(v_5, v_6) \in S(v_1, v_2)$, implying that S contains two conflicting edges, (ii) or an edge (w_3, w_4) such that $(w_3, w_4) \oplus (u_7, u_8)$, with $(u_7, u_8) \in S(u_2, u_3)$ implying that S contains two conflicting edges, (iii) or an edge (w_3, w_4) such that $(w_3, w_4) \oplus (u_3, u_4)$, implying that (u_3, u_4) conflicts with two candidate saturating edges, (iv) or an edge (w_3, w_4) such that $(w_3, w_4) \oplus (v_1, v_2)$, implying that (v_1, v_2) conflicts with two candidate saturating edges.

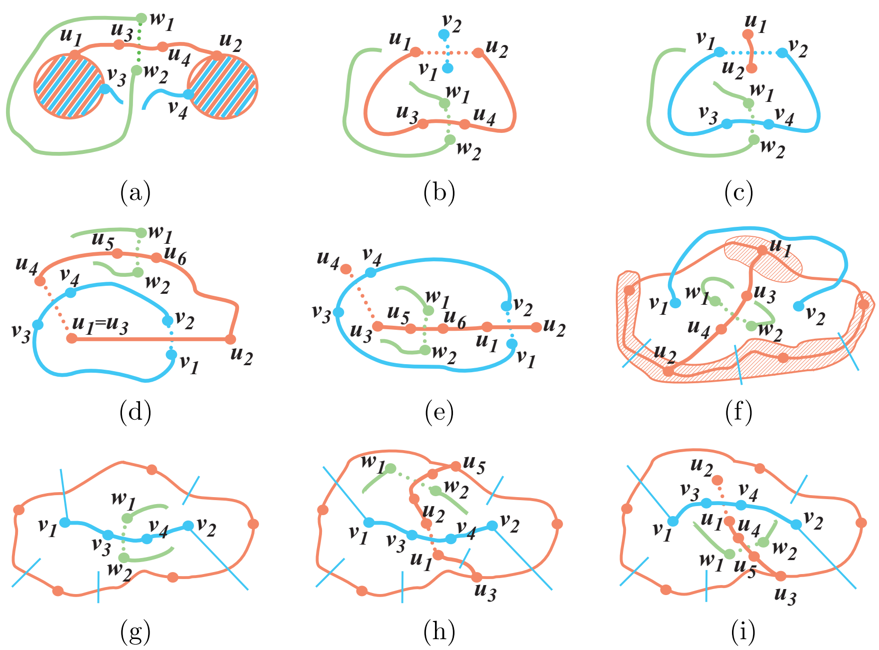


Figure 10: Illustrations for the necessity of the conditions of Theorem 2. Edges of $\mathcal{G}_i^*$ are red, edges of $\mathcal{G}_j^*$ are light blue, and edges of $\mathcal{G}_k^*$ are green.

Now we prove the necessity of Condition 5. By definition, each edge of $\mathcal{G}_{i,j}^*$ has a conflict with (and hence is dual to) one edge of $\mathcal{G}_{j,i}^*$ and vice versa. Moreover, by the necessity of Condition 4, we can assume that both $\mathcal{G}_{i,j}^*$ and $\mathcal{G}_{j,i}^*$ are edge 2-connected. Hence $\mathcal{G}_{i,j}^*$ is not dual to $\mathcal{G}_{j,i}^*$ only if there is a face of $\mathcal{G}_{i,j}^*$ that contains in its interior two vertices of $\mathcal{G}_{j,i}^*$, or vice versa. Suppose w.l.o.g. that a face f of $\mathcal{G}_{i,j}^*$ contains in its interior two vertices v_1 and v_2 of $\mathcal{G}_{j,i}^*$. Suppose, for a contradiction, that a saturator S of C exists. Consider path

$S(v_1, v_2)$. Fig. 11 shows the strategy of the proof of such a contradiction.

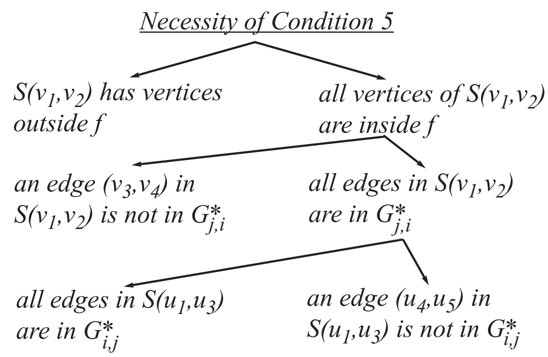


Figure 11: Proof of the necessity of Condition 5. Vertices v_1 and v_2 are both in face f. If all edges of $S(v_1, v_2)$ belong to $\mathcal{G}^*_{j,i}$, then (u_1, u_2) is an edge that has a conflict with an edge of $S(v_1, v_2)$. Vertex u_3 is in f.

- If $S(v_1, v_2)$ is composed in part by vertices inside f and in part by vertices outside f (see Fig. 10.f), consider two vertices u_1 and u_2 in different connected components, disconnected by $S(v_1, v_2)$, of f. Consider $S(u_1, u_2)$. There exists an edge $(u_3, u_4) \in S(u_1, u_2)$ such that $(u_3, u_4) \oplus (w_1, w_2)$, where $(w_1, w_2) \in \mathcal{G}^*_k$, with $k \neq i, j$, otherwise f would not be a face. Hence, $S(w_1, w_2)$ either contains an edge (w_3, w_4) such that $(w_3, w_4) \oplus (u_5, u_6)$, with $(u_5, u_6) \in S(u_1, u_2)$, implying that S contains two conflicting edges, or contains an edge (w_3, w_4) conflicting with an edge $(u_5, u_6) \in f$, implying that (u_5, u_6) conflicts with two candidate saturating edges.

- Otherwise, $S(v_1, v_2)$ is composed by vertices all lying inside f.

  - If there exists an edge $(v_3, v_4) \in S(v_1, v_2)$ such that $(v_3, v_4) \oplus (w_1, w_2)$, where $(w_1, w_2) \in \mathcal{G}^*_k$, with $k \neq i, j$ (see Fig. 10.g), then $S(w_1, w_2)$ contains: (i) either an edge (w_3, w_4) such that $(w_3, w_4) \oplus (v_5, v_6)$, with $(v_5, v_6) \in S(v_1, v_2)$, implying that S contains two conflicting edges, (ii) or an edge (w_3, w_4) such that $(w_3, w_4) \oplus (u_1, u_2)$, with $(u_1, u_2) \in f$, implying that (u_1, u_2) conflicts with two candidate saturating edges, (iii) or an edge (w_3, w_4) such that $(w_3, w_4) \oplus (v_5, v_6)$, with (v_5, v_6) dual to an edge of f, implying that (v_5, v_6) conflicts with two candidate saturating edges.
  - Otherwise, each edge of $S(v_1, v_2)$ is dual to an edge of $\mathcal{G}^*_{i,j}$. Consider any edge (u_1, u_2) dual to an edge of $S(v_1, v_2)$.
    - If $u_1 \in f$ or if there exists a vertex $u_3 \in f$ such that all edges of $S(u_1, u_3)$ conflict with edges of $\mathcal{G}^*_{j,i}$ (see Fig. 10.h), then $u_2 \notin f$

and there exists no vertex u_4 in f such that all edges of $S(u_2, u_4)$ conflict with edges of $\mathcal{G}^*_{j,i}$, otherwise f would not be a face. Consider any vertex $u_5 \in f$ and path $S(u_2, u_5)$. Then, there exists an edge in $S(u_2, u_5)$ that has a conflict with an edge (w_1, w_2) in $\mathcal{G}^*_k$, with $k \neq i, j$. Hence, path $S(w_1, w_2)$ contains: (i) either an edge (w_3, w_4) such that $(w_3, w_4) \oplus (v_5, v_6)$, with $(v_5, v_6) \in S(v_1, v_2)$, implying that S contains two conflicting edges, (ii) or an edge (w_3, w_4) such that $(w_3, w_4) \oplus (u_6, u_7)$, with $(u_6, u_7) \in S(u_2, u_5)$, implying that S contains two conflicting edges, (iii) or an edge (w_3, w_4) such that $(w_3, w_4) \oplus (u_6, u_7)$, with $(u_6, u_7) \in f$, implying that (u_6, u_7) conflicts with two candidate saturating edges, (iv) or an edge (w_3, w_4) such that $(w_3, w_4) \oplus (v_5, v_6)$, with (v_5, v_6) dual to an edge in f, implying that (v_5, v_6) conflicts with two candidate saturating edges.

∗ If $u_1 \notin f$ and there exists no vertex $u_3 \in f$ such that every edge of $S(u_1, u_3)$ conflicts with an edge of $\mathcal{G}^*_{j,i}$ (see Fig. 10.i), then there exists a vertex $u_3 \in f$ such that $S(u_1, u_3)$ contains an edge (u_4, u_5) such that $(u_4, u_5) \oplus (w_1, w_2)$, with $(w_1, w_2) \in \mathcal{G}^*_k$, with $k \neq i, j$, and a contradiction is derived as in the previous case.

Now we prove the sufficiency of Conditions 1, 2, 3, 4, and 5, for the c-planarity of $C(G, T)$. Consider any planar drawing of G satisfying Conditions 1 and 2 and hence satisfying Conditions 1 and 2 of Theorem 1. We show how to construct an acyclic saturator S of C satisfying Condition 3 of Theorem 1. Apply the simplification phase. As a result, get an acyclic set E of edges already chosen to be in S and a candidate saturating graph $\mathcal{G}^*_i$ for each cluster μ_i (this can be done since C satisfies Condition 3).

Order the clusters in whichever way $\mu_1, \mu_2, \dots, \mu_m$. For any pair of clusters μ_i and μ_j, with $i < j$, choose a spanning tree $\mathcal{T}^*_{i,j}$ of $\mathcal{G}^*_{i,j}$ (a spanning tree of $\mathcal{G}^*_{i,j}$ can always be found since, by Condition 4, $\mathcal{G}^*_{i,j}$ is edge 2-connected). Remove from $\mathcal{G}^*_{j,i}$ all edges dual to edges of $\mathcal{T}^*_{i,j}$, obtaining a graph $\mathcal{T}^*_{j,i}$. We claim that $\mathcal{T}^*_{j,i}$ is a spanning tree of $\mathcal{G}^*_{j,i}$. By Condition 5, $\mathcal{G}^*_{i,j}$ and $\mathcal{G}^*_{j,i}$ are dual graphs, and, since they are edge 2-connected, the edges of a cycle in $\mathcal{G}^*_{i,j}$ are dual to the edges of a cutset in $\mathcal{G}^*_{j,i}$, and vice versa (Lemma 1.4 of [14]). Hence, if $\mathcal{T}^*_{j,i}$ has more than one connected component, then the edges removed from $\mathcal{G}^*_{j,i}$ form a cutset for $\mathcal{G}^*_{j,i}$, and the edges of $\mathcal{T}^*_{i,j}$ form a cycle, contradicting the hypothesis that $\mathcal{T}^*_{i,j}$ is a tree. Moreover, if a set of edges of $\mathcal{T}^*_{j,i}$ is a cycle, then the edges dual to such a cycle form a cutset for $\mathcal{G}^*_{i,j}$, contradicting the hypothesis that $\mathcal{T}^*_{i,j}$ is spanning for $\mathcal{G}^*_{i,j}$.

For any pair of clusters μ_i and μ_j, with $i < j$, add the edges of $\mathcal{T}^*_{i,j}$ and the edges of $\mathcal{T}^*_{j,i}$ to S. We claim that S is an acyclic saturator of C. Namely, we prove that (1) no two edges of S have a conflict, (2) adding the edges of S to G connects the subgraph induced by each cluster, and (3) adding the edges of S to G does not create cycles composed by vertices all belonging to the same cluster.

1. *No two edges of S have a conflict:* Edges chosen in the simplification phase

do not conflict each other by construction. Such edges do not conflict with edges of trees $\mathcal{T}^*_{i,j}$. In fact, an edge in $\mathcal{T}^*_{i,j}$ conflicts only with an edge in $\mathcal{G}^*_j$, with $i \neq j$. By construction, edges of the $\mathcal{T}^*_{i,j}$'s do not conflict each other.

2. *Adding the edges of S to G connects the subgraph induced by each cluster:* Distinct connected components of $G(\mu_i)$ are represented after the simplification phase by distinct vertices in $\mathcal{G}^*_i$, that is edge 2-connected and that is partitioned in edge 2-connected subgraphs $\mathcal{G}^*_{i,j}$. Since a spanning tree is chosen to be in S for any $\mathcal{G}^*_{i,j}$, we have that $\bigcup_j \mathcal{T}^*_{i,j}$ is spanning for $\mathcal{G}^*_i$ and $G'(\mu_i)$ has exactly one connected component. Recall that $G'(\mu_i)$ is the graph obtained by adding the edges of the saturator to $G(\mu_i)$.

3. *Adding the edges of S to G does not create cycles composed by vertices all belonging to the same cluster:* Suppose that $G'(\mu_i)$ has a cycle $\mathcal{C}$ containing an edge of S. By construction, edges chosen in the simplification phase only join different connected components of $G(\mu_i)$ and no edge of $\mathcal{C}$ could belong to some $\mathcal{G}^*_{i,j}$, otherwise $G'(\mu_j)$ would be disconnected.

Since S is an acyclic saturator of C, the conditions of Theorem 1 are satisfied by C, that hence is c-planar. □

Theorem 3 *An embedded flat clustered graph $C(G,T)$ with at most five vertices per face is c-planar if and only if:*

1. *G is planar;*

2. *There exists a face f in G such that when f is chosen as outer face for G no cycle composed by vertices of the same cluster encloses a vertex of a different cluster;*

3. *Each cluster of C admits a candidate saturating graph;*

4. *For each pair of distinct clusters μ_i and μ_j, $\mathcal{G}^*_{i,j}$ is edge 2-connected; and*

5. *For each pair of distinct clusters μ_i and μ_j, $\mathcal{G}^*_{i,j}$ is dual to $\mathcal{G}^*_{j,i}$.*

Proof: Consider any face f of G. Since f has at most five vertices, it has at most two connected components of each cluster, so it has at most one candidate saturating edge for each cluster. Since at least two vertices are necessary for each candidate saturating edge, f contains candidate saturating edges for at most two clusters. Hence, C is a single-conflict embedded flat clustered graph and the statement follows from Theorem 2. □

4 An Efficient c-Planarity Testing Algorithm

In this section we use Theorem 3 to derive a linear time and space c-planarity testing algorithm for embedded flat clustered graphs with at most five vertices per face. The algorithm can be extended to test the c-planarity of single-conflict embedded flat clustered graphs relying on Theorem 2. However, it turns out that such an extension exploits several technicalities, in order to deal with a number of candidate saturating edges that can be asymptotically more than linear in the number of vertices of the clustered graph. Hence, to improve the readability of the section, we give the algorithm for the case of embedded flat clustered graphs with at most five vertices per face, while emphasizing the steps of the algorithm that have to be modified to deal with single-conflict embedded flat clustered graphs.

Let $C(G,T)$ be an n-vertex embedded flat clustered graph with at most five vertices per face. To test Condition 1 of Theorem 3, it is sufficient to test if G is a planar embedding. This can be done in $O(n)$ time and space with the techniques in [13].

To test Condition 2, we observe that a face exists satisfying such a condition if and only if the embedded clustered graph is *hole-free*, that is, chosen an arbitrary face as external, there exists no cycle $\mathcal{C}$ that is composed by vertices of the same cluster μ and that separates two vertices both belonging to clusters different from μ (see Fig. 12).

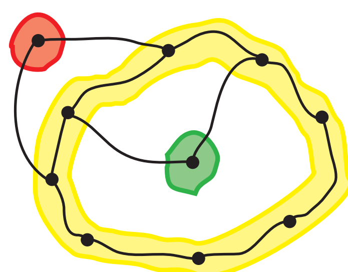


Figure 12: A hole in an embedded clustered graph. A hole consists of a cycle that is composed by vertices of the same cluster μ and that separates two vertices both belonging to clusters different from μ. An embedded clustered graph having no hole is said to be *hole-free*.

A linear-time algorithm for checking if an embedded clustered graph is hole-free has been provided in [6] in the case of c-connected clustered graphs. We can use the same algorithm because of the following lemma.

Lemma 7 *Let $C(G,T)$ be an embedded clustered graph. Let $C'(G,T')$ be the embedded c-connected clustered graph obtained from C as follows. Each node ν of T is replaced in T' by nodes $\nu_1,\dots,\nu_h$, one for each of the $h \geq 1$ connected components of $G(\nu)$. Let $\mu_1,\dots,\mu_k$ be the nodes replacing the parent μ of ν. The parent of ν_j in T' is the node μ_i such that $G(\nu_j)$ is a subgraph of $G(\mu_i)$. We have that C is hole-free if and only if C' is hole-free.*

Proof: Suppose that C is hole-free and suppose, for a contradiction, that C' is not hole-free. Choose arbitrarily in G an external face. Then, a cycle $\mathcal{C}$ of G

exists composed by vertices of the same cluster $\mu_i \in T'$ such that $\mathcal{C}$ has a vertex v_1 inside and a vertex v_2 outside both belonging to clusters in T' different from μ_i. Consider cluster $\mu \in T$ that is replaced in T' by a set of clusters among which there is μ_i. By construction the vertices of $\mathcal{C}$ belong to μ in C.

We claim that there exists a vertex inside $\mathcal{C}$ that does not belong to μ. Since $v_1 \notin \mu_i$, there are two cases: Either $v_1 \notin \mu$, or $v_1 \in \mu$. In the first case the claim directly follows. In the second case, since v_1 and $\mathcal{C}$ belong to μ but are in different clusters in C', they belong to different connected components of $G(\mu)$. Consider any path internal to $\mathcal{C}$ connecting v_1 to a vertex of $\mathcal{C}$. Such a path exists, otherwise G would not be connected. The vertices of such a path cannot all belong to μ, otherwise v_1 and $\mathcal{C}$ would be in the same connected component of $G(\mu)$. Hence, there exists a vertex internal to $\mathcal{C}$ not belonging to μ and the claim follows. A similar argument proves that there exists a vertex outside $\mathcal{C}$ that does not belong to μ, that gives the desired contradiction.

Now suppose C' is hole-free and suppose, for a contradiction, that C is not hole-free. Choose arbitrarily in G an external face. Then, a cycle $\mathcal{C}$ of G exists composed by vertices of the same cluster $\mu \in T$ such that $\mathcal{C}$ has a vertex v_1 inside and a vertex v_2 outside both belonging to clusters in T different from μ. Then, consider cluster μ_i containing $\mathcal{C}$ in C'. Since $v_1, v_2 \notin \mu$, we have that $v_1, v_2 \notin \mu_i$, that gives the desired contradiction. □

In order to test Condition 3 we need to create multigraphs $\mathcal{G}_i$. This is done in $O(n)$ time as follows.

- *Connected Components.* For each node μ of T compute the connected components of $G(\mu)$. This is easily done in linear time and space. Each vertex v of $G(\mu)$ is labelled by a name uniquely associated with the connected component of $G(\mu)$ vertex v belongs to.

- *Candidate saturating edges.* We insert candidate saturating edges inside the faces of G. Consider a face f. Construct maximal sequences of vertices that are consecutive on the border of f and that belong to the same cluster. For any two sequences V_1 and V_2 that have vertices belonging to the same cluster μ, take a vertex $v_1 \in V_1$ and a vertex $v_2 \in V_2$. Test in constant time if the connected component $G_i(\mu)$ of $G(\mu)$ labelling v_1 is different from the connected component $G_j(\mu)$ labelling v_2. If $G_i(\mu)$ is not the same connected component of $G_j(\mu)$, then insert a candidate saturating edge between v_1 and v_2. As already discussed in the proof of Theorem 3, at most two edges are inserted inside f. Since f has at most five vertices, the described insertion can be performed in constant time and hence in linear time for all the faces of G.

  This step is more tricky when considering single-conflict clustered graphs, where faces can have a linear number of vertices. In that case, in order to achieve linear time special care must be taken when considering groups of candidate saturating edges between vertices of the same cluster and when determining the conflicts between candidate saturating edges.

Namely, consider a face f and a cluster μ having connected components $G_1(\mu), \dots, G_k(\mu)$ in f.

If $k = 1$ no candidate saturating edge is inserted in f.

If $k > 2$ (see Fig. 13.a), then we insert in f one candidate saturating edge between any vertex of $G_i(\mu)$ and any vertex of $G_{i+1}(\mu)$, for $i = 1, \dots, k-1$. In fact, in this case, since C is single-conflict, none of such edges has a conflict with any other candidate saturating edge e (otherwise e would have more than one conflict). Hence, since such edges are conflict-free no other edge is required in order to connect the components of cluster μ in f.

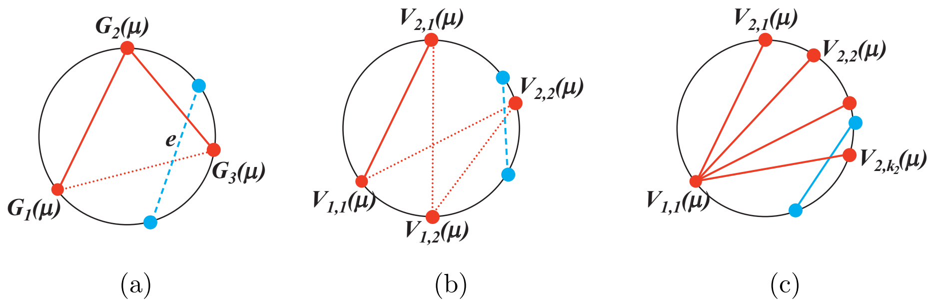


Figure 13: Candidate saturating edges for a cluster μ_i in a single-conflict clustered graph: (a) $k > 2$. (b) $k = 2$, $k_1 > 1$, and $k_2 > 1$. (c) $k = 2$ and $k_1 = 1$. The candidate saturating edges that are added to the graphs $\mathcal{G}_i$'s are solid. Dashed cyan edges correspond to candidate saturating edges that cannot exist, otherwise the clustered graph would not be single-conflict. Dashed red edges correspond to candidate saturating edges that are not needed to connect the components of μ_i in f.

Suppose that $k = 2$ and let $V_{1,1}(\mu), \dots, V_{1,k_1}(\mu)$ ($V_{2,1}(\mu), \dots, V_{2,k_2}(\mu)$) be the maximal sequences of vertices that are consecutive on the border of f and that belong to $G_1(\mu)$ (resp. to $G_2(\mu)$).

If both $k_1 > 1$ and $k_2 > 1$ (see Fig. 13.b), we add a candidate saturating edge between any vertex of $V_{1,1}(\mu)$ and any vertex of $V_{2,1}(\mu)$. Such an edge is conflict-free and we can repeat the above arguments to show that no other edge is required to connect the components of μ in f.

If either $k_1 = 1$ or $k_2 = 1$, say $k_1 = 1$ (see Fig. 13.c), we add edges between any vertex of $V_{1,1}(\mu)$ and any vertex of $V_{2,i}(\mu)$ (with $i = 1, \dots, k_2$). Observe that such edges might have conflicts.

By the above discussion, the number of candidate saturating edges inserted for each cluster μ inside f is linear in the number of maximal sequences of vertices that are consecutive on the border of f and that belong to $G(\mu)$. It follows that a total linear number of candidate saturating edges are inserted into the faces of G. Further, such edges are sufficient to find a saturator for the clustered graph, if any such a saturator exists.

At this point we detect conflicts. We traverse the border of f in clockwise direction starting at any vertex. During the traversal, we maintain a list of encountered edges in a stack P. At each encountered vertex v we do what follows: We consider the candidate saturating edges incident to v in clockwise order; for each edge e, if e has never been encountered we insert e into P; otherwise, the first end-vertex of e has already been encountered and e is already in P. We check if e has a conflict with the top edge e' of P. If yes, we record the conflict and remove e and e' from P. If not, we remove e from P. Such a procedure detects all conflicts among candidate saturating edges. In fact, the conflict structure of the candidate saturating edges is parenthetic, due to the restriction to single-conflict clustered graphs.

- *Multigraphs* $\mathcal{G}_i$. Consider cluster μ_i. Add a vertex to $\mathcal{G}_i$ for each connected component of $G(\mu_i)$. For each of the above mentioned candidate saturating edges (u, v), insert an edge between the connected components of u and v. For each edge e in a multigraph $\mathcal{G}_i$, we record the edge e^* that has a conflict with e, if any. The construction of the $\mathcal{G}_i$'s can be done such that their embeddings are those induced by the adjacencies of the faces of G. Further, such a construction can be done in linear time and space because of the following:

  Property 6 *$\sum_{\mu_i} |\mathcal{G}_i| = O(n)$, where $|\mathcal{G}_i|$ is the size of the graph.*

  Proof: The total number of vertices of the $\mathcal{G}_i$'s is at most the number of vertices of G, hence it is bounded by n.

  If each face of G has at most five vertices the proof is trivial. In fact, there are at most two candidate saturating edges for each face. Hence, the total number of edges of the $\mathcal{G}_i$'s is $O(n)$.

  On the other hand, when considering single-conflict embedded flat clustered graphs, that can generally have faces with a linear number of incident vertices, we apply the algorithm described above, that inserts for each face f only a number of edges that is linear in the size of f. □

Now we show how to test if Condition 3 of Theorem 3 is satisfied. First, test if the $\mathcal{G}_i$'s are connected. If not, return non-c-planar.

We equip each $\mathcal{G}_i$ with a data structure supporting the following update operations, which are trivial graph operations and that can be hence performed in constant time: Remove an edge, collapse (identify the end-vertices of) an edge and merge the embeddings of its end-vertices. Observe the difference between the above definition of the collapse operation and the one given in Section 3, where the edges between the end-vertices are removed.

Next, we show how to apply the simplification phase. We first deal with conflict-free edges, that are edges with no conflict, and we will apply Simplification 1 till the multigraphs $\mathcal{G}_i$'s have no conflict-free edge. Second, we will handle separating edges by either applying Simplification 2 till the multigraphs $\mathcal{G}_i$'s have no separating edge or the non-c-planarity of C has been established.

- *Conflict-free edges.* Extract from all $\mathcal{G}_i$'s the candidate saturating edges that have no conflict. Insert all such edges into a set called $\mathcal{F}$. For each edge e of $\mathcal{F}$ compute the set $\mathcal{E}$ of edges parallel to e. Such computations are easily performed in linear time.

  Construct the set $\mathcal{F}'$ of the edges of any spanning forest of $\mathcal{F}$. Let $\mathcal{F}''$ be the set containing the edges that have no conflict after the edges of $\mathcal{F}'$ have been collapsed. We construct $\mathcal{F}''$ as follows. Initialize $\mathcal{F}'' = \emptyset$. Take each edge e_1 of $\mathcal{F}'$. Consider the set $\mathcal{E}$ of edges parallel to e_1. For each edge $e_2 \neq e_1$ in $\mathcal{E}$, if e_2 has a conflict with an edge e_2^*, add e_2^* to $\mathcal{F}''$. After this work has been performed on all the edges of $\mathcal{F}'$, collapse all of such edges, removing self-loops. We have the following:

  Lemma 8 *The edges of set $\mathcal{F}''$ do not have parallel edges.*

  Proof: Suppose, for a contradiction, that after Simplification 1 has been performed on all edges of $\mathcal{F}'$, $\mathcal{F}''$ contains an edge $e_1 \in \mathcal{G}_i$ joining vertices u and v, such that there exists an edge $e_2 \in \mathcal{G}_i$ also joining vertices u and v. Since $e_1 \in \mathcal{F}''$, there exists an edge e_3 joining vertices w and x that has been removed when applying Simplification 1 to collapse an edge e_4 also joining vertices w and x. Consider the step s_i of Simplification 1 in which e_4 has been collapsed. Since e_3 cannot have a conflict with both e_1 and e_2, vertices w and x are before step s_i one inside and one outside the cycle composed of the edges e_1 and e_2 (see Fig. 14). Hence, before step s_i, e_4 either intersects e_1 or e_2, that gives us a contradiction, since e_4 is supposed to be a conflict-free edge, otherwise it would have not been collapsed during an application of Simplification 1. □

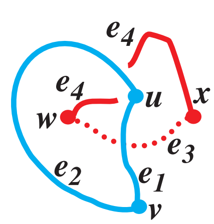
  

  Figure 14: Illustration for the proof of Lemma 8

  Compute any spanning forest of the edges of $\mathcal{F}''$ and perform Simplification 1 on all the edges of such a forest. The above lemma guarantees that after this second pass no new conflict-free edge can be originated.

- *Separating edges.* After the end of the previous step, a set of current multigraphs $\mathcal{G}_i$'s is returned. Exploiting such multigraphs, a set $\mathcal{H}$ of separating edges is constructed as follows. First, associate a name to each face of each multigraph $\mathcal{G}_i$. Second, for each edge e in each multigraph $\mathcal{G}_i$, record the names of the two faces incident to e. Third, for each edge e in each multigraph $\mathcal{G}_i$, verify if the faces incident to e are the same. If yes, then add e to $\mathcal{H}$. Observe that $\mathcal{H}$ is a set containing edges coming from all $\mathcal{G}_i$'s. Each edge e is labelled with a value indicating that e is a separating edge. This computation takes time linear in the number of edges in the $\mathcal{G}_i$'s.

  After the set $\mathcal{H}$ has been created, for each edge e in $\mathcal{H}$, check if the edge e^* conflicting with e is a separating edge. If yes, return non-c-planar. Otherwise, delete e^* and collapse e. Observe that e has no parallel edges, otherwise it would not be a separating edge. After this has been done for all edges in $\mathcal{H}$, it is easy to see that no conflict-free edge has been created. On the other hand, some edges in $\mathcal{G}_i$ could now be separating edges. However, if this happens, then we can conclude that C is not c-planar as stated in the following lemmas:

Lemma 9 *Consider a face f of $\mathcal{G}_i$. Suppose that f contains a separating pair composed by edges (u_1, u_2) and (u_3, u_4). Suppose that (u_1, u_2) has a conflict with an edge (v_1, v_2) that is a separating edge, and that (u_3, u_4) has a conflict with an edge (v_3, v_4). We have that C is not c-planar.*

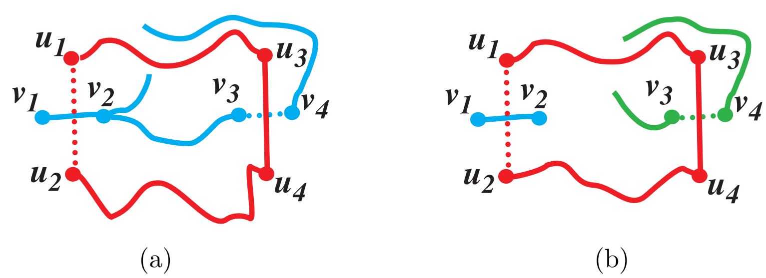


Figure 15: Illustrations for the proof of Lemma 9

Proof: Suppose w.l.o.g. that $(v_1, v_2) \in \mathcal{G}_j$ and that removing (u_1, u_2) and (u_3, u_4) disconnects $\mathcal{G}_i$ in two connected components $\mathcal{G}_i^1$ and $\mathcal{G}_i^2$ such that $u_1, u_3 \in \mathcal{G}_i^1$ and $u_2, u_4 \in \mathcal{G}_i^2$. By Lemma 5, C admits a saturator only if it admits an acyclic saturator S such that $(v_1, v_2) \in S$ and $(u_1, u_2) \notin S$. Since (u_1, u_2) and (u_3, u_4) compose a separating pair, $(u_3, u_4) \in S$, otherwise no path in S could connect $\mathcal{G}_i^1$ and $\mathcal{G}_i^2$. Consider paths $S(u_1, u_3)$ and $S(u_2, u_4)$ connecting u_1 and u_3, and connecting u_2 and u_4 in S (such paths are single vertices if $u_1 = u_3$ and/or $u_2 = u_4$).

If $(v_3, v_4) \in \mathcal{G}_j$ (see Fig. 15.a), then let v_1 (v_2) be the one out of v_1 and v_2 that is outside (resp. inside) f and let v_4 (v_3) be the one out of v_3 and v_4 that is outside (resp. inside) f. Then, v_3 (v_4) is inside (resp. outside) cycle $\mathcal{C} = S(u_1, u_3) \cup (u_3, u_4) \cup S(u_2, u_4) \cup (u_1, u_2)$. Consider path $S(v_2, v_3)$ connecting v_2 and v_3 in S. Notice that $S(v_2, v_3)$ lies completely inside $\mathcal{C}$, otherwise $S(v_2, v_3)$ would contain an edge conflicting with an edge of $S(u_1, u_3)$, or an edge conflicting with an edge of $S(u_2, u_4)$, or an edge conflicting with (u_3, u_4), implying that S contains two conflicting edges. Consider path $S(v_2, v_4)$ connecting v_2 and v_4 in S. Since v_2 is inside $\mathcal{C}$ and v_4 is outside $\mathcal{C}$, $S(v_2, v_4)$ lies in part inside and in part outside $\mathcal{C}$. It follows that either there exists an edge of $S(v_2, v_4)$ conflicting with an edge of $S(u_1, u_3)$, or an edge conflicting with an edge of $S(u_2, u_4)$, or an edge conflicting with (u_3, u_4), implying that S contains two conflicting edges, or $S(v_2, v_4)$ contains edge (v_1, v_2). However, since (v_1, v_2) is a separating edge no path excluding (v_1, v_2) and connecting v_1 to v_4 exists in $\mathcal{G}_i$.

If $(v_3, v_4) \in \mathcal{G}_k$, with $k \neq i, j$, then vertices v_3 and v_4 lie one inside and one outside $\mathcal{C}$. Hence, any path connecting v_3 and v_4 in S either contains an edge conflicting with an edge of $S(u_1, u_3)$, or with an edge of $S(u_2, u_4)$, or with (u_3, u_4) implying that S contains two conflicting edges, or contains an edge conflicting with (u_1, u_2), implying that (u_1, u_2) has two conflicting edges, respectively. □

Lemma 10 *Suppose that each edge of $\mathcal{H}$ has a conflict with a non-separating edge. Collapse the edges in $\mathcal{H}$, repeatedly applying Simplification 2. Either the resulting multigraphs $\mathcal{G}_i$ are edge 2-connected or C is not c-planar.*

Proof: Order the edges of $\mathcal{H}$ in whichever way $\{e_1, e_2, \ldots, e_k\}$. Let e_j, with $1 \leq j \leq k$, be the first edge in $\{e_1, e_2, \ldots, e_k\}$ such that (i) collapsing edges $e_1, e_2, \ldots, e_{j-1}$ from the $\mathcal{G}_i$'s does not create new separating edges and (ii) collapsing edge e_j creates a new separating edge. Suppose, for a contradiction, that a saturator of C exists. Then, by Lemma 5, there exists a saturator S containing edges $e_1, e_2, \ldots, e_{j-1}$ and not containing the edges that have conflicts with edges $e_1, e_2, \ldots, e_{j-1}$. Consider the $\mathcal{G}_i$'s after edges $e_1, e_2, \ldots, e_{j-1}$ have been collapsed (and the edges that have conflicts with edges $e_1, e_2, \ldots, e_{j-1}$ have been removed). Refer to Fig. 15. Since collapsing edge $e_j = (v_1, v_2)$ creates a new separating edge (u_3, u_4), (u_1, u_2) and (u_3, u_4) compose a separation pair for a multigraph $\mathcal{G}_i$, where (u_1, u_2) is the edge that has a conflict with (v_1, v_2). Hence, there exists a face of $\mathcal{G}_i$ containing (u_1, u_2) and (u_3, u_4). Since no edge (and hence neither (u_3, u_4)) is conflict-free, the statement follows from Lemma 9. □

After the collapse of all the edges in $\mathcal{H}$ and the removal of their conflicting edges, a set of current multigraphs $\mathcal{G}_i$'s is returned. Exploiting such multigraphs, Condition 3 can be tested as follows. First, associate a name to each face of each multigraph $\mathcal{G}_i$; second, for each edge e in each multigraph $\mathcal{G}_i$, record the names of the two faces incident to e, and third, for each edge e in each multigraph $\mathcal{G}_i$, verify if the faces incident to e are the same. If this is true for at least one edge, by the previous lemmas we can return that the input graph is not c-planar, otherwise the current $\mathcal{G}_i$'s are the candidate saturating graphs of the clusters.

For each pair of distinct clusters μ_i and μ_j, we check if $\mathcal{G}^*_{i,j}$ is edge 2-connected (Condition 4 of Theorem 3) and if $\mathcal{G}^*_{i,j}$ is dual to $\mathcal{G}^*_{j,i}$ (Condition 5 of Theorem 3). This is easily done in linear time because of the following property.

Property 7 $\sum_{i,j} |\mathcal{G}^*_{i,j}| = O(n)$*, where* $|\mathcal{G}^*_{i,j}|$ *is the size of the graph.*

Proof: It trivially follows from Property 6. □

Hence, we can conclude the section with the following theorem.

Theorem 4 *The c-planarity of an n-vertex embedded flat clustered graph* $C(G,T)$ *with at most five vertices per face can be tested in* $O(n)$ *time and space.*

As a consequence of the arguments discussed in this section, we remark that a theorem analogous to Theorem 4 holds even for single-conflict clustered graphs.

5 Conclusions

In this paper we have shown that the c-planarity of embedded flat clustered graphs with at most five vertices per face and, more generally, the c-planarity of single-conflict embedded flat clustered graphs can be efficiently tested.

We remark that the simplification phase described in Section 3 is a preprocessing that can be performed on any embedded flat clustered graph. This allows to reduce the problem of testing the c-planarity of such graphs to the one of deciding whether a set of edge 2-connected candidate saturating graphs admits a set of non-conflicting spanning trees. However, it's rather easy to see that the characterization shown in Theorem 2 does not hold for general embedded flat clustered graphs.

We conclude by providing a list of families of embedded clustered graphs for which, in our opinion, determining the time complexity of a c-planarity testing is worth of interest: (i) single-conflict general (non-flat) embedded clustered graphs; (ii) embedded flat clustered graphs such that for each face of the underlying graph there are at most two (or a constant number of) vertices of the same cluster; and (iii) embedded flat clustered graphs.

References

[1] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(6):476–497, 1994.

[2] S. Cornelsen and D. Wagner. Completely connected clustered graphs. *Journal of Discrete Algorithms*, 4(2):313–323, 2006.

[3] P. F. Cortese and G. Di Battista. Clustered planarity. In J. S. B. Mitchell and G. Rote, editors, *Proc. Symposium on Computational Geometry*, pages 32–34, 2005.

[4] P. F. Cortese, G. Di Battista, M. Patrignani, and M. Pizzonia. On embedding a cycle in a plane graph. In P. Healy and N. S. Nikolov, editors, *Proc. Graph Drawing*, volume 3843 of *LNCS*, pages 49–60, 2006.

[5] P. F. Cortese, G. Di Battista, M. Patrignani, and M. Pizzonia. Clustering cycles into cycles of clusters. *Journal of Graph Algorithms and Applications*, 9(3):391–413, 2005.

[6] E. Dahlhaus. A linear time algorithm to recognize clustered graphs and its parallelization. In C. L. Lucchesi and A. V. Moura, editors, *Proc. Latin American Theoretical Informatics*, volume 1380 of *LNCS*, pages 239–248, 1998.

[7] G. Di Battista and F. Frati. Efficient c-planarity testing for embedded flat clustered graphs with small faces. In S.-H. Hong, T. Nishizeki, and W. Quan, editors, *Proc. Graph Drawing*, volume 4875 of *LNCS*, pages 291–302, 2008.

[8] Q. Feng, R. F. Cohen, and P. Eades. Planarity for clustered graphs. In P. G. Spirakis, editor, *Proc. European Symposium on Algorithms*, volume 979 of *LNCS*, pages 213–226, 1995.

[9] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001.

[10] M. T. Goodrich, G. S. Lueker, and J. Z. Sun. C-planarity of extrovert clustered graphs. In P. Healy and N. S. Nikolov, editors, *Proc. Graph Drawing*, volume 3843 of *LNCS*, pages 211–222, 2006.

[11] C. Gutwenger, M. Jünger, S. Leipert, P. Mutzel, M. Percan, and R. Weiskircher. Advances in c-planarity testing of clustered graphs. In S. G. Kobourov and M. T. Goodrich, editors, *Proc. Graph Drawing*, volume 2528 of *LNCS*, pages 220–235, 2002.

[12] E. Jelinkova, J. Kara, J. Kratochvil, M. Pergel, O. Suchy, and T. Vyskocil. Clustered planarity: Small clusters in eulerian graphs. In S.-H. Hong, T. Nishizeki, and W. Quan, editors, *Proc. Graph Drawing*, volume 4875 of *LNCS*, pages 303–314, 2008.

[13] D. G. Kirkpatrick. Establishing order in planar subdivisions. *Discrete & Computational Geometry*, 3:267–280, 1988.

[14] T. Nishizeki and N. Chiba. *Planar Graphs: Theory and Algorithms*. North-Holland, 1988.

[15] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.