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

Giuseppe Di Battista and Fabrizio Frati

Dipartimento di Informatica e Automazione – Università di Roma Tre

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 shades new light on the c-planarity testing problem.

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 open.

A brief survey on the problem of testing the c-planarity of clustered graphs can be found in [2]. 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 algorithm for this class has been presented in [7]. *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 [1]. *Almost connected clustered graphs*, in which either all nodes corresponding to non-connected clusters are in the same path in the cluster hierarchy, or for each non-connected cluster its parent and all its siblings are connected [9]. *Extrovert clustered graphs*, a generalization of c-connected clustered graphs with special restrictions on the cluster hierarchy [8]. *Cycles of clusters*, in which the hierarchy is *flat*, the underlying graph is a simple cycle, and the clusters are arranged in a cycle [3]. 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 a 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 NP-hard Graph Drawing problems

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

S.-H. Hong, T. Nishizeki, and W. Quan (Eds.): GD 2007, LNCS 4875, pp. 291–302, 2007.

on planar graphs that become polynomial-time solvable if the embedding is fixed. Testing if a graph admits an orthogonal planar drawing with at most k bends or if a graph admits an upward planar drawing 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 of the embedded underlying graph with extra edges such that the resulting graph is c-connected and c-planar. We call *candidate saturating edges* those edges that are 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 and an efficient c-planarity testing algorithm 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. Characterization and algorithm for clustered graphs with at most five vertices per face are a consequence of such a more general result.

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 test. Section 5 contains conclusions and open problems. Because of space limits some proofs are in the full version of the paper [6].

2 Preliminaries

A graph G is *vertex (edge) k-connected* if the removal of any $k-1$ vertices (edges) leaves G connected. A *separating edge* 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* 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 planar 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 in the border of the face.

A *clustered graph* is a pair $C(G,T)$, where G is a planar graph and T is a rooted tree whose leaves 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 $V(\mu)$ (called *cluster*) of the vertices of G that are leaves of the subtree of T rooted at μ; $G(\mu)$

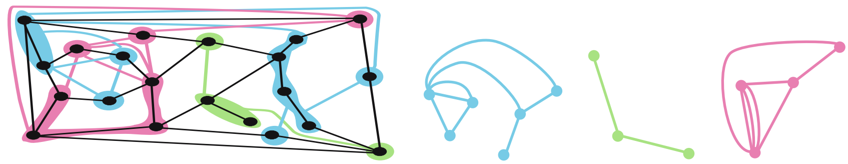

Fig. 1. (a) An embedded flat clustered graph C and its candidate saturating edges. Different clusters have different colors. (b)–(c)–(d) Multigraphs $\mathcal{G}_i$ for C.

denotes the subgraph of G induced by the vertices in $V(\mu)$. If each cluster induces a connected subgraph of G, then C is *c-connected*, otherwise C is non-c-connected. An *embedded clustered graph* is a clustered graph such that G is connected and the planar embedding of the underlying graph of C is fixed. A *flat clustered graph* is such that the number of nodes in any path from the root to a leaf of T is three. 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.

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 don't 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. 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.

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 $V_i \neq V_j$ 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 are edges that can be added to the clustered graph to make it c-connected (see Fig. 1). 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. The embedding of $\mathcal{G}_i$ is given by the order of the faces around the vertices of G. 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 and e_2, joining connected components $G_1(\mu_i)$ and $G_2(\mu_i)$ of $G(\mu_i)$, and $G_1(\mu_j)$ and $G_2(\mu_j)$ of $G(\mu_j)$, respectively, 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, 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 candidate saturating edges for the c-planarity of a flat embedded clustered graph. Even if not explicitly stated, Theorem 1 has been used in [3].

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 each cluster induces in G' exactly one connected component.*

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. 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.*

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

3 A Characterization

We restrict ourselves to embedded flat clustered graphs in which each candidate saturating edge has a conflict with at most one other candidate saturating edge. We call *single-conflict* an embedded flat clustered graph satisfying such a property. Consider a single-conflict embedded flat clustered graph $C(G, T)$ and the multigraph $\mathcal{G}_i$ for each cluster μ_i in T. We have the following structural lemma.

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.*

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

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

There are edges in the $\mathcal{G}_i$'s that must be used in any saturator of C and edges that are not 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 $\mathcal{G}_i$.

The operation of *removing* an edge e from $\mathcal{G}_i$ corresponds to the choice of not using e in 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 possibly 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 in the saturator of C. It consists of: (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 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 in v but not in 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.

The algorithm 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 $\mathcal{G}_i$ 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 $\mathcal{G}_i$ 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 $\mathcal{G}_i$ 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 $\mathcal{G}_i$ *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*. The following properties can be easily proved.

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

Property 2. The subgraphs induced by the collapsed edges are acyclic.

Property 3. Candidate saturating graphs are planar embedded and edge 2-connected.

Property 4. Any edge of a candidate saturating graph has exactly one conflict with an edge of a different candidate saturating graph.

We now 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. s_m can be one of Simplification 1, 2, or 3. Denote by $s_0, s_1, \ldots, s_{m-1}$ the simplifications performed before s_m and denote by E the set of edges collapsed while applying $s_0, s_1, \ldots, s_{m-1}$. Inductively, suppose that if an acyclic saturator of C exists, there exists an acyclic saturator composed by 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 and 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. 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, then 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 and 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 $\mathcal{G}_i$, then no path composed by edges of E connects u and v. Since e_i is a separating edge, then 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 in S, then $e_j \notin S$. □

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 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 $\mathcal{G}_j$), then no path composed by edges of E connects u and v (connects w and x). Since e_i and e_j are separating edges, then 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:

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: Let S be an acyclic saturator of C and let u and v be two vertices of candidate saturating graph $\mathcal{G}_i^*$. Denote by $S(u, v)$ the path of S connecting u and v. If edges e_i and e_j of 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 one 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. Denote by v_1 and v_2 vertices in different connected components. Suppose, for a contradiction, that an acyclic saturator S of C exists. Consider $S(v_1, v_2)$ (see Fig. 2.a). Since v_1 and v_2 are in different components of $\mathcal{G}_{i,j}^*$, there exists an edge $(u, v) \in S(v_1, v_2)$ s. t. $(u, v) \oplus (w, x)$, where $(w, x) \in \mathcal{G}_k^*$, with $k \neq i, j$. Consider $S(w, x)$. Each edge of $S(w, x)$ cannot have a conflict with any edge of $S(v_1, v_2)$, otherwise S would contain two conflicting edges, and with any edge e of $\mathcal{G}_{i,j}^*$, otherwise e would conflict with two candidate saturating edges. Hence, $\mathcal{G}^*_{j,i}$ has at least two connected components. Let u_1 and u_2 be two vertices in such components, respectively. Then, $S(u_1, u_2)$ either contains an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(v_1, v_2)$, or contains an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(w, x)$, implying that S contains two conflicting edges.

Now suppose that $\mathcal{G}^*_{i,j}$ has a separating edge (u, v). By construction $(u, v) \oplus (w, x)$, where $(w, x) \in \mathcal{G}^*_{j,i}$. Suppose, for a contradiction, that a saturator S of C exists.

1. If $(u, v) \notin S$, then consider $S(u, v)$ (see Fig. 2.b). Since (u, v) is a separating edge for $\mathcal{G}^*_{i,j}$, then there exists an edge $(u', v') \in S(u, v)$ s. t. $(u', v') \oplus (w', x')$, where $(w', x') \in \mathcal{G}^*_k$, with $k \neq i, j$. Hence, $S(w', x')$ either contains an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(u, v)$, implying that S contains two conflicting edges, or contains an edge e_1 conflicting with (u, v), implying that (u, v) conflicts with two candidate saturating edges.
2. If $(u, v) \in S$, then consider $S(w, x)$.
   - If an edge $(w', x') \in S(w, x)$ is s. t. $(w', x') \oplus (u', v')$, with $(u', v') \notin \mathcal{G}^*_{i,j}$, a contradiction is obtained as in the previous case (see Fig. 2.c).
   - Otherwise, consider any edge $(w', x') \in S(w, x)$ and edge $(u', v') \in \mathcal{G}^*_{i,j}$ s. t. $(u', v') \oplus (w', x')$. Let v (v') be the endpoint of (u, v) (resp. of (u', v')) outside cycle $S(w, x) \cup (w, x)$.
     - If $u = u'$ or if all edges of $S(u, u')$ have conflicts with edges of $\mathcal{G}^*_{j,i}$ (see Fig. 2.d), consider $S(v, v')$. Then there exists an edge $(u'', v'') \in S(v, v')$ s. t. $(u'', v'') \oplus (w'', x'')$, where $(w'', x'') \in \mathcal{G}^*_k$, with $k \neq i, j$, otherwise (u, v) would not be a separating edge. Hence, $S(w'', x'')$ either contains an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(v, v')$, implying that S contains two conflicting edges, or an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(u, u')$, implying that S contains two conflicting edges, or an edge e_1 s. t. $e_1 \oplus (u', v')$, implying that (u', v') conflicts with two candidate saturating edges, or an edge e_1 s. t. $e_1 \oplus (u, v)$, implying that (u, v) conflicts with two candidate saturating edges.
     - If $u \neq u'$ and $S(u, u')$ contains at least one edge (u'', v'') s. t. $(u'', v'') \oplus (w'', x'')$, where $(w'', x'') \in \mathcal{G}^*_k$, with $k \neq i, j$ (see Fig. 2.e), then $S(w'', x'')$ contains either an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(w, x)$, implying that S contains two conflicting edges, or an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(u, u')$, or an edge e_1 s. t. $e_1 \oplus (u', v')$, implying that (u', v') conflicts with two candidate saturating edges, or an edge e_1 s. t. $e_1 \oplus (w, x)$, implying that (w, x) conflicts with two candidate saturating edges.

Now we prove the necessity of Condition (5). 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. 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 u and v of $\mathcal{G}^*_{j,i}$. Suppose, for a contradiction, that a saturator S of C exists. Consider $S(u, v)$.

1. If the vertices of $S(u, v)$ are in part inside f and in part outside f (see Fig. 2.f), consider two vertices v_1 and v_2 in different connected components, disconnected by $S(u, v)$, of f. Consider $S(v_1, v_2)$. There exists an edge $(w, x) \in S(v_1, v_2)$ s. t.

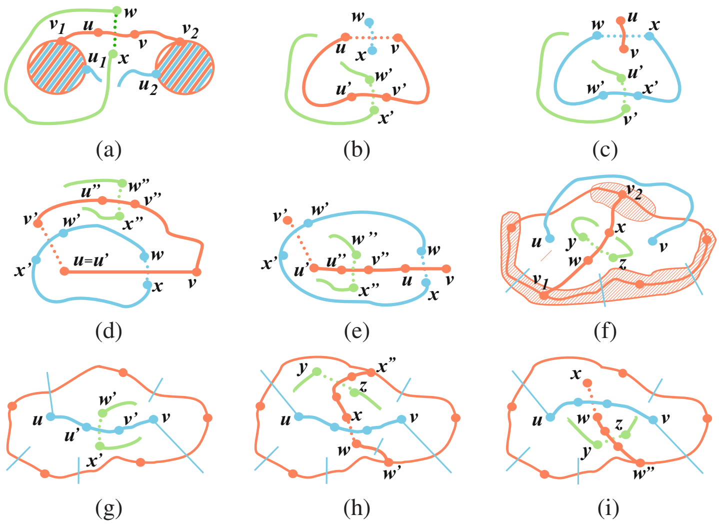

Fig. 2. Illustrations for the necessity of the conditions of Theorem 2. Edges of $\mathcal{G}_i^*$ are red, edges of $\mathcal{G}_j^*$ are light blue, and edges of $\mathcal{G}_k^*$ are green.

$(w,x) \oplus (y,z)$, where $(y,z) \in \mathcal{G}_k^*$, with $k \neq i,j$, otherwise f would not be a face. Hence, $S(y,z)$ either contains an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(v_1,v_2)$, implying that S contains two conflicting edges, or an edge e_1 conflicting with an edge $e_2 \in f$, implying that e_2 conflicts with two candidate saturating edges.

2. Otherwise, $S(u,v)$ is composed by vertices all lying inside f.
   - If there is an edge $(u',v') \in S(u,v)$ s. t. $(u',v') \oplus (w',x')$, where $(w',x') \in \mathcal{G}_k^*$, with $k \neq i,j$ (see Fig. 2.g), then $S(w',x')$ either contains an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(u,v)$, implying that S contains two conflicting edges, or an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in f$, implying that e_2 conflicts with two candidate saturating edges, or an edge e_1 s. t. $e_1 \oplus e_2$, with e_2 dual to an edge of f, implying that e_2 conflicts with two candidate saturating edges.
   - Otherwise, any edge of $S(u,v)$ is dual to an edge of $\mathcal{G}_{i,j}^*$. Consider any edge (w,x) dual to an edge of $S(u,v)$.
     - If $w \in f$ or if there exists a vertex $w' \in f$ s. t. any edge of $S(w,w')$ conflicts with an edge of $\mathcal{G}_{j,i}^*$ (see Fig. 2.h), then $x \notin f$ and there exists no vertex x' in f s. t. all edges of $S(x,x')$ conflict with edges of $\mathcal{G}_{j,i}^*$, otherwise f would not be a face. Consider any vertex $x'' \in f$ and $S(x,x'')$. Then, there exists an edge in $S(x,x'')$ that has a conflict with an edge (y,z) in $\mathcal{G}_k^*$, with $k \neq i,j$. Hence, $S(y,z)$ either contains an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(u,v)$, implying that S contains two conflicting edges, or contains an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in S(x,x'')$, implying that S contains two conflicting edges, or contains an edge e_1 s. t. $e_1 \oplus e_2$, with $e_2 \in f$, implying that e_2 conflicts with two candidate saturating edges, or contains an edge e_1 s. t. $e_1 \oplus e_2$, with e_2 dual to an edge in f, implying that e_2 conflicts with two candidate saturating edges.

- If $w \notin f$ and there exists no vertex $w' \in f$ s. t. any edge of $S(w, w')$ conflicts with an edge of $\mathcal{G}^*_{j,i}$ (see Fig. 2.i), then there exists a vertex $w'' \in f$ s. t. $S(w, w'')$ contains an edge e_1 s. t. $e_1 \oplus (y, z)$, with $(y, z) \in \mathcal{G}^*_k$, with $k \neq i, j$, and a contradiction is derived as in the previous case.

We prove the sufficiency of Conditions 1, 2, 3, 4, and 5 for the c-planarity of C. 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, choosing an acyclic set E of edges to be in S and obtaining 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 $\mu_1, \mu_2, \ldots, \mu_m$. For any pair of clusters μ_i and μ_j, with $i < j$, choose a spanning tree $\mathcal{T}^*_{i,j}$ of $\mathcal{G}^*_{i,j}$ ($\mathcal{T}^*_{i,j}$ can 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 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 [11]). Hence, if $\mathcal{T}^*_{j,i}$ has more than one connected component, the edges removed from $\mathcal{G}^*_{j,i}$ form a cutset for $\mathcal{G}^*_{j,i}$, and those of $\mathcal{T}^*_{i,j}$ form a cycle, contradicting the hypothesis that $\mathcal{T}^*_{i,j}$ is a tree. If a set of edges of $\mathcal{T}^*_{j,i}$ is a cycle, 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 of $\mathcal{T}^*_{j,i}$ to S. We claim that S is a saturator of C. 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. Hence, S does not have two conflicting edges. It's easy to see that, after G has been augmented to a graph G' by adding the edges of S to it, each cluster μ_i has exactly one connected component. Namely, 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 $\mathcal{G}^*_{i,j}$, then $\bigcup_j \mathcal{T}^*_{i,j}$ is spanning for $\mathcal{G}^*_i$ and $G'(\mu_i)$ has exactly one connected component. Finally, suppose that $G'(\mu_i)$ has a cycle 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 c could belong to some $\mathcal{G}^*_{i,j}$, otherwise $G'(\mu_j)$ would be disconnected. □

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; and (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, then it has at most two connected components of each cluster, so it has at most one candidate saturating edge per cluster. Since at least two vertices are necessary for each candidate saturating edge, then f contains candidate saturating edges for at most two clusters. Hence, C is a single-conflict embedded flat clustered graph and Theorem 2 applies. □

4 An Efficient c-Planarity Testing Algorithm

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. The details of the extension are omitted for brevity. Anyway, we will emphasize the steps of the algorithm that have to be modified for this purpose.

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 [10].

To test Condition (2), we observe that a face exists satisfying such a condition iff the embedded clustered graph is *hole-free*, that is, chosen an arbitrary face as external, a cycle c of G doesn't exist composed by vertices of the same cluster μ such that c has a vertex inside and a vertex outside both belonging to clusters different from μ. A linear-time algorithm for checking if an embedded clustered graph is hole-free has been provided in [5] in the case of c-connected clustered graphs. However, we can use the same algorithm because of the following lemma.

Lemma 7. *Let $C(G,T)$ be a clustered graph. Let $C'(G,T')$ be the 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 iff C' is hole-free.*

In order to test Condition (3) we 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 done in linear time and space. **Candidate saturating edges.** We insert candidate saturating edges inside the faces of G. Consider a face f. Construct maximal sequences of vertices consecutive on the border of f and belonging to the same cluster. For any two sequences S_1 and S_2 that have vertices belonging to the same cluster, take a vertex $v_1 \in S_1$ and a vertex $v_2 \in S_2$. If the connected component associated to v_1 is different from the one associated to v_2 (this can be tested in constant time), then insert a candidate saturating edge between v_1 and v_2. At most two edges are inserted inside f. The described insertion can be performed in constant time and hence in linear time for all faces of G. This step is more tricky when considering single-conflict clustered graphs. In this case, in order to achieve total linear time special care must be put when considering groups of candidate saturating edges between vertices of the same cluster and when determining the conflicts between candidate saturating edges. **Multigraphs $\mathcal{G}_\mathbf{i}$.** Consider cluster μ_i. Add a vertex to $\mathcal{G}_i$ for each connected component of $G(\mu_i)$. For each candidate saturating edge e insert an edge between the connected components joined by e. The construction of the $\mathcal{G}_i$'s can be done so 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 5. $\sum_{\mu_i} |\mathcal{G}_i| = O(n)$, where $|\mathcal{G}_i|$ is the size of the graph.

Property 5 does not hold when considering single-conflict embedded flat clustered graphs, that can generally have faces with a linear number of incident vertices. However, the arrangement of the candidate saturating edges in the single-conflict setting allows to reduce the size of the construction introducing only an overall linear number of them.

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 and query operations: remove an edge, collapse (identify the end-vertices of) an edge and merge the embeddings of its end-vertices, answer if an edge is a separating edge, answer if an edge has a conflict and in case output the conflicting edge. Observe the difference between the above definition of the collapse operation and the one given in Section 3. A data structure exists that can be set-up in linear time and that performs each of the above operations in constant time. In fact, all of them are trivial graph operations, with the exception of answering if an edge e is a separating edge. We equip each edge with two pointers to the two identifiers of the incident faces. When an edge e is removed from $\mathcal{G}_i$ we simply modify the identifier of one of the two faces former incident on e. To answer the query in constant time we check if the two faces around e are the same face. Also, we compute a set $\mathcal{F}$ of candidate saturating edges that have no conflict. For each edge e of $\mathcal{F}$ we compute the set $\mathcal{E}$ of edges parallel to e. Such computations are performed in linear time. We will show how to use $\mathcal{F}$ and sets $\mathcal{E}$ during the simplification steps.

We show how to apply Simplification 1. Construct the set $\mathcal{F}'$ of the edges of any spanning forest of $\mathcal{F}$. Set $\mathcal{F}'' = \emptyset$. 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. Set $\mathcal{F}''$ contains all the edges that became conflict-free after the previous step. The edges of $\mathcal{F}''$ do not have multiple edges:

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

Perform Simplification 1 on the edges of $\mathcal{F}''$. The above lemma guarantees that after this second pass no new conflict-free edge can be originated.

Now, Simplification 2 is applied till the $\mathcal{G}_i$'s are edge 2-connected or the non-c-planarity of C is stated. First, construct a set $\mathcal{H}$ of separating edges as follows. For each edge e in $\mathcal{G}_i$ verify if the faces incident to e are the same. If yes, then add e to $\mathcal{H}$. This computation takes time linear in the number of edges in the $\mathcal{G}_i$'s. Now, for each edge e in $\mathcal{H}$, check if edge e^* conflicting with e belongs to $\mathcal{H}$. If yes, return non-c-planar, otherwise delete e^* and collapse e. After this has been done for all edges in $\mathcal{H}$, other separating edges could have been created in $\mathcal{G}_i$. 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 edge (v_1, v_2) that is a separating edge, and that (u_3, u_4) has a conflict with edge (v_3, v_4). We have that C is not c-planar.*

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.*

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

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.*

5 Conclusions

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 where each face of the underlying graph has at most two (or a constant number of) vertices of the same cluster; and (iii) embedded flat clustered graphs.

References

1. Cornelsen, S., Wagner, D.: Completely connected clustered graphs. Journal of Discrete Algorithms 4(2), 313–323 (2006)
2. Cortese, P.F., Di Battista, G.: Clustered planarity. In: ACM SoCG 2005, pp. 32–34 (2005)
3. Cortese, P.F., Di Battista, G., Patrignani, M., Pizzonia, M.: Clustering cycles into cycles of clusters. Journal of Graph Algorithms and Applications 9(3), 391–413 (2005)
4. Cortese, P.F., Di Battista, G., Patrignani, M., Pizzonia, M.: On embedding a cycle in a plane graph. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 49–60. Springer, Heidelberg (2006)
5. Dahlhaus, E.: A linear time algorithm to recognize clustered graphs and its parallelization. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN 1998. LNCS, vol. 1380, pp. 239–248. Springer, Heidelberg (1998)
6. Di Battista, G., Frati, F.: Efficient c-planarity testing for embedded flat clustered graphs with small faces. Tech. Report RT-DIA-119-2007, Dip. Inf. e Aut., Univ. Roma Tre (2007)
7. Feng, Q., Cohen, R.F., Eades, P.: Planarity for clustered graphs. In: Spirakis, P.G. (ed.) ESA 1995. LNCS, vol. 979, pp. 213–226. Springer, Heidelberg (1995)
8. Goodrich, M.T., Lueker, G.S., Sun, J.Z.: C-planarity of extrovert clustered graphs. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 211–222. Springer, Heidelberg (2006)
9. Gutwenger, C., Jünger, M., Leipert, S., Mutzel, P., Percan, M., Weiskircher, R.: Advances in c-planarity testing of clustered graphs. In: Goodrich, M.T., Kobourov, S.G. (eds.) GD 2002. LNCS, vol. 2528, pp. 220–235. Springer, Heidelberg (2002)
10. Kirkpatrick, D.G.: Establishing order in planar subdivisions. Discrete & Computational Geometry 3, 267–280 (1988)
11. Nishizeki, T., Chiba, N.: Planar Graphs: Theory and Algorithms. North-Holland (1988)