Clustered Planarity: Embedded Clustered Graphs with Two-Component Clusters

Vít Jelínek[1⋆], Eva Jelínková[1], Jan Kratochvíl[1,2], and Bernard Lidický[1]

[1] Department of Applied Mathematics[⋆⋆]
[2] Institute for Theoretical Computer Science[⋆⋆⋆]
Charles University
Malostranské nám. 25, 118 00 Praha, Czech Republic
`{jelinek,eva,honza,bernard}@kam.mff.cuni.cz`

Abstract. We present a polynomial-time algorithm for c-planarity testing of clustered graphs with fixed plane embedding and such that every cluster induces a subgraph with at most two connected components.

1 Introduction

Clustered planarity (or shortly, c-planarity) has recently become an intensively studied topic in the area of graph and network visualization. In many situations one needs to visualize a complicated inner structure of graphs and networks. Clustered graphs provide a possible model of such a visualization, and as such they find applications in many practical problems, e.g., management information systems, social networks or VLSI design tools [5]. However, from the theoretical point of view, the computational complexity of deciding c-planarity is still an open problem and it is regarded as one of the challenges of contemporary graph drawing.

A *clustered graph* is a pair $(G, \mathcal{C})$, where $G = (V, E)$ is a graph and $\mathcal{C}$ is a family of subsets of V (called *clusters*), with the property that each two clusters are either disjoint or in inclusion. We always assume that the vertex set V is in $\mathcal{C}$, and we call it *the root cluster*. We say that a clustered graph $(G, \mathcal{C})$ is *clustered-planar* (or shortly *c-planar*), if the graph G has a planar drawing such that we may assign to every cluster $X \in \mathcal{C}$ a compact simply connected region of the plane which contains precisely the vertices of X and whose boundary crosses every edge of G at most once (see Sect. 2 for the precise definition).

It is well known that planar graphs can be recognized in polynomial (even linear) time. For c-planarity, determining the time-complexity of the decision problem remains open; only partial results are known. If every cluster of $(G, \mathcal{C})$ induces a connected subgraph of G, then the c-planarity of $(G, \mathcal{C})$ can be tested in linear time by an algorithm of Dahlhaus [3], which improves upon a polynomial algorithm of Feng et al. [5]. Several generalizations of this result are known:

⋆ Supported by the grant 201/05/H014 of the Czech Science Foundation.
⋆⋆ Supported by project MSM0021620838 of the Czech Ministry of Education.
⋆⋆⋆ Supported by grant 1M0545 of the Czech Ministry of Education.

c-planarity testing is polynomial for clustered graphs in which all disconnected clusters form a single chain in the cluster hierarchy [7], for clustered graphs in which for every disconnected cluster X, the parent cluster and all the sibling clusters of X are connected [7], and for clustered graphs where every disconnected cluster X has connected parent cluster, with the additional assumption that each component of X is adjacent to a vertex not belonging to the parent of X [6].

Another approach to c-planarity testing is to consider *flat clustered graphs*, which are clustered graphs in which all non-root clusters are disjoint. Even in this restricted setting, the complexity of c-planarity testing is unknown. However, polynomial-time algorithms exist for special types of flat clustered graphs, e.g., if the underlying graph is a cycle and the clusters are arranged in a cycle [2], if the underlying graph is a cycle and the clusters are arranged into an embedded plane graph [1], or if the underlying graph is a cycle and the clusters contain at most three vertices [9]. Even for these very restricted settings, the algorithms are quite non-trivial.

Suppose an embedding of the underlying graph is fixed. Does the c-planarity testing become easier? This question was already addressed in [4], who provide a linear algorithm for flat clustered graphs with a prescribed embedding in which all faces have size at most five.

In this paper, we also deal with clustered graphs $(G, \mathcal{C})$, for which the embedding of G is fixed. In this setting, we obtain a polynomial algorithm for c-planarity of clustered graphs in which each cluster induces a subgraph with at most two connected components.

Theorem 1. *There is a polynomial time algorithm for deciding c-planarity of a clustered graph $(G, \mathcal{C})$, where G is a plane graph and every cluster of $\mathcal{C}$ induces a subgraph of G with at most two connected components.*

In the paper, we present a simplified version of the algorithm which assumes that the cluster hierarchy is flat. The general algorithm for non-flat clustered graphs is described in the Appendix. The proofs of our lemmas are moved to the Appendix as well.

2 Preliminaries

We follow standard terminology on finite simple loopless plane graphs. A *plane graph* is an ordered pair $G = (V, E)$, where V is a finite set of points in the plane (called *vertices*) and E is a set of Jordan arcs (called *edges*), such that every edge connects two distinct vertices of G and avoids any other vertex, every pair of vertices is connected by at most one edge, and no two edges intersect, except in a possible common endpoint.

If $G = (V, E)$ is a plane graph and $X \subseteq V$ is a set of vertices, we let $\overline{X}$ denote the set $V \setminus X$ and we let $G[X]$ denote the subgraph of G induced by X.

Two plane graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* if there is a continuous bijection f of the plane with continuous inverse such that $V' = \{f(v) \colon v \in V\}$ and $E' = \{f[e] \colon e \in E\}$ (where $f[e]$ is the set $\{f(x) \colon x \in e\}$).

The algorithm we will present in this paper expects a representation of a plane graph as part of its input. Since the algorithm does not need to make a distinction between isomorphic plane graphs, we may represent a plane graph G by a data structure which identifies G uniquely up to isomorphism. We may identify the isomorphism class of G by specifying, for every vertex of G, the cyclic order of edges and faces incident to v, and by specifying the outer face of G. The isomorphism class of a plane graph can be thus represented by a data structure whose size is polynomial in $|V|$.

Let $G = (V, E)$ be a plane graph. A *cluster set* on G is a set $\mathcal{C} \subseteq \mathcal{P}(V(G))$ such that for all $X, Y \in \mathcal{C}$, either X and Y are disjoint or they are in inclusion; the pair $(G, \mathcal{C})$ is called a *plane clustered graph*. The elements of $\mathcal{C}$ are called *clusters*. We assume that the set $V(G)$ is always in $\mathcal{C}$, and we call it the *root cluster*. A cluster that does not contain any other cluster as a subset is called *minimal*.

Clusters are naturally ordered by inclusion. The set $V(G)$ is the maximum of this ordering. A cluster is called *connected* if it induces in G a connected subgraph and *disconnected* otherwise. A *component* of a cluster $X \in \mathcal{C}$ is a maximal set $X_1 \subseteq X$ such that $G[X_1]$ is a connected subgraph of $G[X]$.

We say that a plane clustered graph $(G, \mathcal{C})$ is *connected* (or *2-connected*, or *disconnected*) if the graph G is connected (or 2-connected, or disconnected). Let us remark that some earlier papers use the term 'connected clustered graph' to denote a clustered graph in which every cluster is connected; we break with this convention for the sake of consistency of our definitions.

In this paper, we consider clustered graphs $(G, \mathcal{C})$ in which every disconnected cluster in $\mathcal{C}$ has exactly two components. We will call such a pair $(G, \mathcal{C})$ a *2-component clustered graph*.

For a plane clustered graph $(G, \mathcal{C})$, a *clustered planar embedding* is a mapping emb_c that assigns to every cluster $X \in \mathcal{C}$ a compact simply connected planar region $emb_c(X)$ (called *the cluster region of* X) whose boundary $\gamma(X)$ is a closed Jordan curve (called *the cluster boundary of* X), such that

- for each vertex $v \in V$ and each cluster $X \in \mathcal{C}$, v is in $emb_c(X)$ if and only if $v \in X$,
- for each cluster $X \in \mathcal{C}$, the cluster boundary $\gamma(X)$ does not contain any vertex from V,
- for every two clusters X and Y, the regions $emb_c(X)$ and $emb_c(Y)$ are disjoint (in inclusion) if and only if X and Y are disjoint (in inclusion, respectively), and
- for every edge $e \in E$ and every cluster $X \in \mathcal{C}$, the edge e crosses the cluster boundary of X at most once.

A plane clustered graph is called *clustered planar* (shortly *c-planar*) if it allows a clustered planar embedding.

When testing c-planarity, we adopt the approach first used in [5] of adding extra edges to the underlying graph in order to make each cluster connected.

Definition 1. *Let $(G, \mathcal{C})$ be a plane clustered graph. Let c be a cycle in G whose vertices all belong to a cluster $X \in \mathcal{C}$. We say that c is a* hole *of the cluster X, if the interior region of c contains a vertex not belonging to X.*

Clearly, a plane clustered graph with a hole is not c-planar. On the other hand, it is known [5] that a plane clustered graph without holes whose clusters are all connected is c-planar. For a given plane clustered graph $(G, \mathcal{C})$ the existence of a hole can be determined in polynomial time [5].

Definition 2. *Let G be a plane graph. A* candidate edge *of G is a simple curve $e \notin E$ such that $(V, E \cup \{e\})$ is a plane graph. A* candidate set *is a set S of candidate edges of G such that $(V, E \cup S)$ is a plane graph. We use the notation $G \cup e$ and $G \cup S$ as a shorthand for $(V, E \cup \{e\})$ and $(V, E \cup S)$ respectively.*

We say that two candidate edges e and e' are isomorphic *if $G \cup e$ and $G \cup e'$ are isomorphic plane graphs.*

Note that a pair of vertices u, v of a plane graph G may be connected by two distinct non-isomorphic candidate edges. On the other hand, it is not hard to see that a plane graph on n vertices has at most $O(n^2)$ non-isomorphic candidate edges.

The following theorem reduces c-planarity testing to searching for a specific set of candidate edges. It was proved in an equivalent version by Feng et al. [5].

Theorem 2. *A plane clustered graph $(G, \mathcal{C})$ is c-planar if and only if there exists a candidate set S with the following properties:*

1. *$(G \cup S, \mathcal{C})$ has no hole,*
2. *every cluster X of $\mathcal{C}$ induces a connected subgraph in $G \cup S$.*

A set S of candidate edges satisfying the above conditions is called a *saturator*[3]. A set S that satisfies the first condition will be called a *partial saturator*. We say that a candidate edge e *saturates* a cluster X, if e connects a pair of vertices belonging to different components of X. A saturator S is *minimal* if no proper subset of S is a saturator. Note that every candidate edge from a minimal saturator S saturates a cluster from $\mathcal{C}$. Moreover, if X is a cluster with two components that does not contain any disconnected subcluster, then a minimal saturator S has exactly one candidate edge saturating X.

Definition 3. *If e is a candidate edge of a plane clustered graph $(G, \mathcal{C})$ such that $(G, \mathcal{C})$ is c-planar if and only if $(G \cup e, \mathcal{C})$ is c-planar, then the edge e is called* harmless. *Similarly, a candidate set S is* harmless *provided $(G, \mathcal{C})$ is c-planar if and only if $(G \cup S, \mathcal{C})$ is c-planar.*

Note that if $(G, \mathcal{C})$ is a c-planar clustered graph, then a candidate set is harmless if and only if it is a subset of a saturator of $(G, \mathcal{C})$. On the other hand, if $(G, \mathcal{C})$ is not c-planar, then any candidate set is harmless.

Let us now present several simple but useful lemmas, whose proof may be found in the Appendix.

[3] Note that this definition of saturator differs slightly from that of some other papers—here, candidate edges are already embedded.

Lemma 1. *Let $(G, \mathcal{C})$ be a plane clustered graph without holes, let $X \in \mathcal{C}$ be a cluster which is minimal and connected. Then $(G, \mathcal{C})$ is c-planar if and only if $(G, \mathcal{C} \setminus \{X\})$ is c-planar.*

The next lemma shows that c-planarity testing of 2-component graphs can be reduced to c-planarity testing of 2-component connected plane clustered graphs.

Lemma 2. *If there is a polynomial time algorithm for deciding c-planarity for connected 2-component plane clustered graphs, then there is a polynomial time algorithm for deciding c-planarity for arbitrary 2-component plane clustered graphs.*

The following lemma allows us to reduce c-planarity testing of a connected graph to an equivalent instance of c-planarity where the underlying graph is 2-connected.

Lemma 3. *Let $(G, \mathcal{C})$ be a connected plane clustered graph with at least three vertices which is not 2-connected. There is a polynomial-time transformation which constructs a plane clustered graph $(G', \mathcal{C}')$ such that G' is connected, G' has fewer components of 2-connectivity than G, $(G', \mathcal{C}')$ is c-planar if and only if $(G, \mathcal{C})$ is c-planar, and there is a bijection f between $\mathcal{C}$ and $\mathcal{C}'$ such that for every cluster $X \in \mathcal{C}$, the graph $G[X]$ has the same number of components as the graph $G'[f(X)]$.*

Thanks to Lemma 3, a connected 2-component plane c-planarity instance $(G, \mathcal{C})$ can be polynomially transformed into an equivalent 2-connected 2-component instance $(G', \mathcal{C}')$. To achieve this, we simply perform repeatedly the transformation described in Lemma 3, until the resulting graph has only one 2-connected component.

Combining Lemma 2 and Lemma 3, we see that to decide the c-planarity of 2-component plane graphs, it is sufficient to provide an algorithm that decides c-planarity of 2-connected 2-component plane graph. This is an important technical simplification, because in a 2-connected plane graph, the boundary of every face is a cycle, and a candidate edge in every inner face is uniquely determined (up to isomorphism) by its end-vertices and the face where it should be drawn.

Unfortunately, if F is the outer face of G, a pair of vertices of F may still be connected by two non-isomorphic candidate edges belonging to F (see Fig. 1). To avoid this technical nuisance, we will restrict the set of candidate edges. Let $(G, \mathcal{C})$ be a 2-connected plane clustered graph, let $f \in E(G)$ be an edge which connects a pair of vertices $u, v \in V(G)$, with the following properties:

- f appears on the boundary of the outer face of G,
- every non-root cluster contains at most one of the two vertices u, v.

Such an edge f exists, otherwise the boundary of the outer face would be a hole of a non-root cluster. We say that a candidate edge e of G is *properly drawn* if f is on the boundary of the outer face of $G \cup e$. Note that every candidate edge in an inner face of G is properly drawn, while a pair of non-adjacent vertices on the boundary of the outer face may be connected by two non-isomorphic candidate edges, exactly one of which is properly drawn. Thus, a properly drawn candidate edge is uniquely determined (up to isomorphism) by its pair of endpoints and the face where it should be embedded.

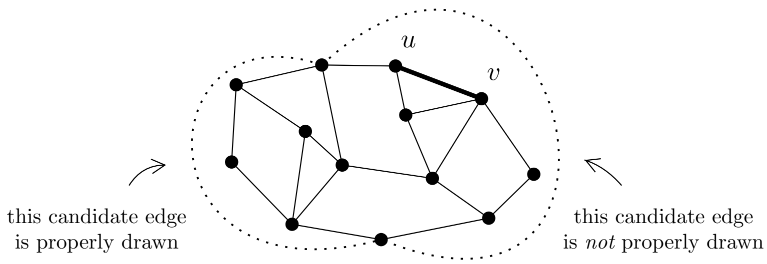

Fig. 1. Two candidate edges connecting the same pair of vertices in the outer face.

It can be shown that if a 2-connected plane clustered graph is c-planar, then it has a saturator that only contains properly drawn candidate edges (see Lemma 4 in the Appendix).

3 The Algorithm

In this section, we present our algorithm deciding the c-planarity of 2-component plane clustered graphs. As mentioned in the introduction, we will only deal with the restricted setting of *flat* clustered graph, i.e., the clustered graphs where all the non-root clusters are minimal. The description of the general case is presented in the Appendix.

Our aim is to find a polynomial algorithm deciding the c-planarity of plane 2-connected 2-component flat clustered graph $(G, \mathcal{C})$.

To achieve this, we will present a polynomial-time procedure FIND-EDGE which, when presented with a 2-component 2-connected hole-free plane clustered graph $(G, \mathcal{C})$ as an input, will either determine that $(G, \mathcal{C})$ is not c-planar, or it will output a harmless candidate edge e that saturates a cluster $X \in \mathcal{C}$. Observe that such a candidate edge e cannot create a hole in $G \cup e$, because both its endpoints belong to different components of X by assumption, and there is no other non-root cluster containing the endpoints of e. This is the main reason why the flat clustered graphs are much easier to deal with than general clustered graphs.

If the procedure FIND-EDGE outputs a harmless candidate edge e, it does not necessarily mean that $(G, \mathcal{C})$ is c-planar. However, since e is harmless, we know that $(G, \mathcal{C})$ is c-planar if and only if $(G \cup e, \mathcal{C})$ is c-planar. We may then call FIND-EDGE again on the input $(G \cup e, \mathcal{C})$, to determine that $(G \cup e, \mathcal{C})$ (and hence also $(G, \mathcal{C})$) is not c-planar, or to find another harmless edge. Since every candidate edge output by the FIND-EDGE procedure saturates a cluster from

$\mathcal{C}$, after at most $|\mathcal{C}|$ invocations of FIND-EDGE we will either obtain a saturator of $(G, \mathcal{C})$ or determine that $(G, \mathcal{C})$ is not c-planar.

The FIND-EDGE algorithm maintains a set P of *permitted edges*. In the beginning, the set P is initialized to contain all the properly drawn candidate edges that saturate a cluster from $\mathcal{C}$. In the first phase of the algorithm, called *the pruning phase*, the algorithm iteratively removes some candidate edges from P, using a set of *pruning rules*, which will be described in Subsection 3.1. The pruning rules guarantee that if $(G, \mathcal{C})$ has a saturator, then it also has a saturator which is a subset of P.

When the set P cannot be further pruned, the algorithm performs the following *triviality checks*, described in detail in Subsection 3.2:

- if there a disconnected cluster that cannot be saturated by any of the permitted edges, then $(G, \mathcal{C})$ is not c-planar,
- if there is a disconnected cluster saturated by a unique permitted edge $e \in P$, then e is harmless,
- if there is a permitted edge e that does not cross any other permitted edge, then e is harmless.

If any of the above conditions is satisfied, the algorithm outputs the corresponding solution and stops. Otherwise, it distinguishes two cases:

1. If there is a disconnected cluster $X \in C$ and a face F of G such that every permitted edge saturating X appears in the face F, then the algorithm performs a subroutine LOCATE-IN-FACE, which will output a harmless permitted edge inside F and stop. This subroutine, together with a brief sketch of its proof, is presented in Subsection 3.3.
2. If the previous case does not apply, it can be shown that any permitted edge is harmless. The algorithm then performs a subroutine called OUTPUT-ANYTHING which outputs an arbitrary permitted edge and stops. The proof of its correctness is sketched in Subsection 3.4.

Before we describe the main parts of the algorithm in greater detail, we need some more terminology.

Let G be a 2-connected plane graph. Let a, b, c, d be a quadruple of distinct vertices on the boundary of a face F of G. We say that the pair ab *crosses* the pair cd in F, if the four vertices appear on the boundary of F in the cyclic order $acbd$. If e and f are two candidate edges of a 2-connected clustered graph $(G, \mathcal{C})$, we say that e *crosses* f if the two candidate edges belong to the same face F of G and the endpoints of e cross with the endpoints of f. For two sets of vertices X and Y, we say that X *crosses* Y *in face* F, if there are vertices $a, b \in X$ and $c, d \in Y$ such that ab crosses cd in the face F.

Most of our arguments rely on the following basic properties of connected subgraphs of 2-connected plane graphs:

- If G is a 2-connected plane graph, and X and Y are disjoint sets of vertices such that $G[X]$ and $G[Y]$ are both connected, then X and Y do not cross in any face of G.

– Let G be a 2-connected plane graph. Let X, Y and Z be disjoint sets of vertices, each of them inducing a connected subgraph of G. Then G has at most two faces that contain vertices of all the three sets on their boundary.

The proof of these properties can be found in the appendix (see Lemma 7 and Lemma 8).

3.1 The Pruning Phase

In the pruning phase, the algorithm FIND-EDGE iteratively restricts the set P of permitted candidate edges. In the beginning of the pruning phase, the set P is initialized to contain all the properly drawn candidate edges that saturate at least one cluster. Note that every permitted edge $e \in P$ saturates a unique cluster $X \in \mathcal{C}$, since we assume that $\mathcal{C}$ is flat. A permitted edge that saturates X will be called an *X-edge*.

If X is a minimal cluster, and if e and e' are two X-edges, we say that e and e' are *equivalent*, if for every permitted edge $f \in P$ that is not an X-edge, the edge f crosses e if and only if it crosses e'.

Throughout the pruning phase, the set P will satisfy the following three invariants.

– For each cluster X and each face F, all the X-edges that belong to F form a vertex-disjoint union of complete bipartite subgraphs; these complete bipartite subgraphs will be called *X-bundles* (or just *bundles*, if X is clear from the context). Two X-edges from different bundles do not cross (see Fig. 2).
– If X and Y are distinct clusters, then if an X-edge e crosses two Y-edges f and f', then f and f' belong to the same bundle.
– If $(G, \mathcal{C})$ is c-planar, then it has a saturator that is a subset of P.

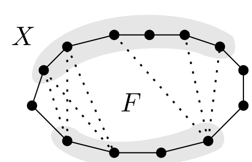

Fig. 2. A face F with two bundles of X-edges.

In the beginning, when P contains all the properly drawn candidate edges that saturate some cluster from C, the three invariants above are satisfied. In fact, if F is a face that contains at least one X-edge, then all the X-edges in F form a complete bipartite graph. Thus, each face has at most one X-bundle.

To prune the set P, we apply the following two rules.

– If, for a cluster X, there is a permitted edge that crosses all the X-edges, then remove from P each edge that crosses all the X-edges.

– Let $e = uv$ and $e' = u'v$ be two X-edges that belong to the same face F and that share a common vertex v. If e and e' are equivalent, remove from P all the X-edges in F incident to u'.

It can be proven that an arbitrary application of one of the rules above preserves all the invariants. The algorithm applies the pruning rules in arbitrary order, reducing the number of permitted edges in each step, until it reaches the situation when none of the rules is applicable. Let us remark that in the general (i.e., non-flat) situation, the pruning is slightly more complicated: there are four pruning rules instead of two, and the rules have assigned priorities which are taken into account when the algorithm selects which rule to apply.

3.2 Triviality Checks

When there is no rule applicable to the set P of permitted edges, the pruning phase ends. The FIND-EDGE algorithm then proceeds with three types of triviality checks, described below.

First, the algorithm checks whether there is a cluster X that is not saturated by any permitted edge. If this is the case, the algorithm concludes that the clustered graph $(G, \mathcal{C})$ is not c-planar and stops. This is a correct conclusion, since if $(G, \mathcal{C})$ were c-planar, then by the last invariant there would have to be a saturator made of permitted edges, which is clearly impossible.

As the next triviality check, the algorithm tries to find a cluster X, such that the set P contains a single X-edge e. If such a cluster X is found, the algorithm outputs e as a harmless edge and stops. This is again a correct output, since by the last invariant, if G is c-planar, then it has a saturator S which is a subset of P. Necessarily, S contains the edge e. This implies that e is harmless.

In the last type of triviality check, the algorithm looks for a permitted edge e that does not cross any permitted edge belonging to a different cluster. If such an edge e is found, the algorithm outputs e as a harmless edge and stops. This is again easily seen to be a correct output.

If none of the triviality checks succeeds, the algorithm counts, for each cluster X, the number of faces of G that contain at least one X-edge. We will say that a cluster X is *one-faced* if all the X-edges belong to a single face of G, X is *two-faced* if all the X-edges appear in the union of two distinct faces, and X is *many-faced* otherwise.

If there is a one-faced cluster X whose permitted edges belong to a face F, then the algorithm performs a subroutine LOCATE-IN-FACE to find a harmless permitted edge in F. This subroutine is described in the next subsection.

If there is no one-faced cluster, it can be shown that all the clusters are two-faced, and that any permitted edge is harmless. The algorithm then outputs an arbitrary permitted edge and stops. The main arguments involved in proving the correctness of this step are sketched in Subsection 3.4.

3.3 LOCATE-IN-FACE

Assume that we are given a set P of permitted edges satisfying all the invariants described in Subsection 3.1. Assume furthermore than none of the pruning rules is applicable to P, and none of the triviality checks has succeeded.

For a face F, we say that a cluster X is an *F-cluster*, if all the X-edges belong to F. We say that a vertex of X is *active*, if it is incident to at least one X-edge.

Assume that F is a face with at least one F-cluster. Using our assumptions about P, we are able to deduce the following facts:

- If X is an F-cluster, and Y is a cluster that has a permitted edge which crosses a permitted edge of X, then Y is also an F-cluster.
- If X is an F-cluster with two components X_1 and X_2, then each component X_i has at most two active vertices. It follows that X has either four permitted edges which all belong to a single bundle, or X has exactly two permitted edges (see Fig. 3; recall that due to the triviality checks, each cluster has at least two permitted edges).

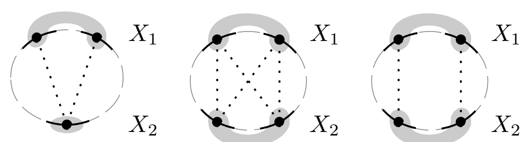

Fig. 3. Possible configurations of permitted edges of an F-cluster X.

Let X be an arbitrary F-cluster, let X_1 and X_2 be its two components. From the triviality checks, we know that every X-edge is crossed by a permitted edge of another cluster. Let $Y \neq X$ be a cluster whose permitted edge crosses an X-edge, and let Y_1 and Y_2 be its two components. Note that a set Y_i may not cross with the set X_j on the boundary of F, because these two sets induce connected subgraphs of G. Recall also, that no Y-edge may intersect all the X-edges (and vice versa), because it would have been pruned.

Putting all these facts together, we conclude that the mutual position of the X-edges and Y-edges corresponds to one of the situations depicted on Fig. 4.

Note that all the configurations of Fig. 4 exhibit a 'mirror symmetry'. To make this observation rigorous, we define a 'symmetry mapping' σ on the set of all the F-active vertices as follows: let X be an arbitrary F-cluster, with components X_1 and X_2. If a component X_i contains two active vertices x and x', then we define $\sigma(x) = x'$ and $\sigma(x') = x$. If X_i contains only one active vertex x, then we put $\sigma(x) = x$. We then extend the mapping σ to the set of X-edges in a natural way: for an X-edge e with endpoints x and y, we define $\sigma(e)$ to be the X-edge with endpoints $\sigma(x)$ and $\sigma(y)$.

The mapping σ has the following properties:

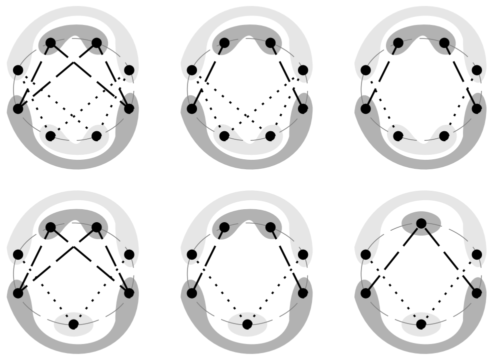

Fig. 4. Mutual positions of permitted edges of two crossing F-clusters.

- For an F-cluster X and an X-edge e, $\sigma(e)$ is an X-edge different from e.
- If X and Y are F-clusters, an X-edge e crosses a Y-edge f if an only if $\sigma(e)$ crosses $\sigma(f)$.
- An X-edge e is harmless if and only if $\sigma(e)$ is harmless.

From these properties, it can be easily deduced that if an F-cluster X has only two permitted edges, then both these edges are harmless.

Furthermore, it is possible to show that if there is at least one F-cluster in a face F, then there is also an F-cluster that has only two permitted edges.

The procedure LOCATE-IN-FACE is then easy to describe: as an input, the procedure expects a face F for which there is at least one F-cluster. The procedure then finds an F-cluster X that has only two permitted edges, and outputs any X-edge as a harmless edge.

3.4 OUTPUT-ANYTHING

If, after the end of the pruning phase, each cluster has permitted edges in at least two distinct faces, and if none of the triviality checks is applicable, we can show that the set P of permitted edges has the following properties:

- For each cluster X, there are exactly two faces of G that contain the X-edges.
- All the X-edges that appear in the same face are equivalent.
- If X and Y are distinct clusters, and if an X-edge crosses a Y-edge, then all the X-edges and all the Y-edges appear in the same pair of faces, and every Y-edge crosses all the X-edges in its face.
- Let $S \subseteq P$ be a minimal saturator of permitted edges. For each edge $e \in S$ find an arbitrary permitted edge $\overline{e}$ that saturates the same cluster as e and appears in a different face than e. The set $\overline{S} = \{\overline{e}\colon e \in S\}$ is another minimal saturator of permitted edges.

From these properties, we may deduce that every permitted edge $e \in P$ is harmless. The procedure OUTPUT-ANYTHING simply outputs an arbitrary permitted edge and stops.

This completes the description of the simplified version of the FIND-EDGE algorithm. It is clear that the algorithm runs in polynomial time.

4 Concluding Remarks

We have shown that c-planarity of 2-component plane clustered graphs can be determined in polynomial time. This result raises several related open problems.

Problem 1. What is the complexity of the c-planarity problem for 2-component graphs $(G, \mathcal{C})$ if the embedding of G is not prescribed?

Problem 2. What is the complexity of deciding the c-planarity of clustered graphs with $O(1)$ components per cluster?

Problem 3. What if we relax the 2-component assumption by allowing the graph G to have arbitrarily many components, and only restricting the number of components of the non-root clusters?

References

1. 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 3843, pp. 49–60, Springer, Heidelberg (2006)
2. Cortese, P.F., Di Battista, G., Patrignani, M., Pizzonia, M.: Clustering cycles into cycles of clusters. Journal of Graph Algorithms and Applications, Special Issue on the 2004 Symposium on Graph Drawing, GD 2004. 9(3):391–413 (2005)
3. Dahlhaus, E.: Linear time algorithm to recognize clustered planar graphs and its parallelization. In: Lucchesi, C. (ed.), LATIN 1998, LNCS, vol. 1380, pp. 239–248, Springer, Heidelberg (1998)
4. Di Battista, G., Frati, F.: Efficient c-planarity testing for embedded flat clustered graphs with small faces. In: Hong, S.-H., Nishizeki, T., Quan, W. (eds.) GD 2007, LNCS, vol. 4875, pp. 291–302, Springer, Heidelberg (2008).
5. Feng, Q.W., Cohen, R.F., Eades, P.: Planarity for clustered graphs. In: Spirakis, P. (ed.) ESA 1995, LNCS, vol. 979, pp. 213–226, Springer-Verlag (1995)
6. 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 3843, pp. 211–222, Springer, Heidelberg (2006)
7. Gutwenger, C., Jünger, M., Leipert, S., Mutzel, P., Percan, M., Weiskircher, R.: Advances in c-planarity testing of clustered graphs. In: Kobourov, S.G., Goodrich, M.T. (eds.) GD 2002, LNCS, vol. 2528, 220–235, Springer, Heidelberg (2002)
8. Gutwenger, C., Jünger, M., Leipert, S., Mutzel, P., Percan, M., Weiskircher, R.: Subgraph induced planar connectivity augmentation. In: Bodlaender, H.L. (ed.) WG 2003, LNCS, vol. 2880, 261–272, Springer, Heidelberg (2003)
9. Jelínková, E., Kára, J., Kratochvíl, J., Pergel, M., Suchý, O., Vyskočil, T.: Clustered planarity: Small clusters in Eulerian graphs. In: Hong, S.-H., Nishizeki, T., Quan, W. (eds.) GD 2007, LNCS, vol. 4875, pp. 303–314, Springer, Heidelberg (2008)

A Appendix

A.1 Omitted Proofs and Other Useful Lemmas

In the first subsection of the appendix we present the omitted proofs from the article as well as other necessary lemmas for more general version of the algorithm. For reader's convenience, we repeat the statements of the lemmas.

Lemma 1. *Let $(G,\mathcal{C})$ be a plane clustered graph without holes, let $X \in \mathcal{C}$ be a cluster which is minimal and connected. Then $(G,\mathcal{C})$ is c-planar if and only if $(G,\mathcal{C} \setminus \{X\})$ is c-planar.*

Proof. Of course, if $(G,\mathcal{C})$ is c-planar then so is $(G,\mathcal{C} \setminus \{X\})$. To prove the other implication, assume that S is a minimal saturator of $(G,\mathcal{C} \setminus \{X\})$. Observe that no two vertices of X are connected by an edge of S, because such a candidate edge does not saturate any cluster of $\mathcal{C} \setminus \{X\}$. In particular, X has no hole in $G \cup S$, which implies that S is a saturator of $(G,\mathcal{C})$. □

Lemma 2. *If there is a polynomial time algorithm for deciding c-planarity for connected 2-component plane clustered graphs, then there is a polynomial time algorithm for deciding c-planarity for arbitrary 2-component plane clustered graphs.*

Proof. Let $(G,\mathcal{C})$ be a disconnected 2-component plane clustered graph. Since the vertex set of G is the root cluster of $\mathcal{C}$, it follows the graph G itself has two connected components.

There are at most $O(n^2)$ non-isomorphic candidate edges that connect two vertices of distinct components of G, and every saturator of the root cluster must contain at least one such edge. Thus, $(G,\mathcal{C})$ is c-planar if and only if there exists a candidate edge e connecting the two components of G such that the connected clustered graph $(G \cup e,\mathcal{C})$ is c-planar. Thus, we can decide c-planarity of $(G,\mathcal{C})$ by $O(n^2)$ invocations of an algorithm deciding c-planarity of connected 2-component plane clustered graphs. □

Lemma 3. *Let $(G,\mathcal{C})$ be a connected plane clustered graph with at least three vertices which is not 2-connected. There is a polynomial-time transformation which constructs a clustered graph $(G',\mathcal{C}')$ such that G' is connected, G' has fewer components of 2-connectivity than G, $(G',\mathcal{C}')$ is c-planar if and only if $(G,\mathcal{C})$ is c-planar, and there is a bijection f between $\mathcal{C}$ and $\mathcal{C}'$ such that for every cluster $X \in \mathcal{C}$, the graph $G[X]$ has the same number of components as the graph $G'[f(X)]$.*

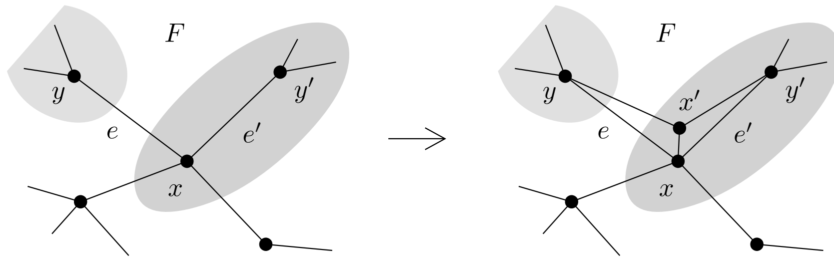

Fig. 5. Decreasing the number of 2-connected components in G.

Proof. See Fig. 5. Let x be a cut-vertex of G. Let $e = xy$ and $e' = xy'$ be two edges of G adjacent to x, such that y belongs to a different 2-connected component of G than y', and the two edges e and e' are consecutive in the cyclic order of edges adjacent to x. Let F be the face of G whose boundary contains the two edges e and e'.

We construct $(G', \mathcal{C}')$ as follows. Let us add to G a new vertex x', embedded in the interior of the face F, and adjacent to the three vertices x, y and y', in such a way that the edge xx' appears between e and e' in the cyclic order of edges incident to x. Let G' be the resulting plane graph. Let us add the vertex x' to all the clusters of $\mathcal{C}$ that contain x, while the remaining clusters are left unchanged. The result of this procedure is illustrated in Fig. 5. Let $\mathcal{C}'$ be the resulting family of clusters.

Note that G' has fewer 2-connected components than G, since the five edges e, e', xx', $x'y$ and $x'y'$ belong to the same 2-connected component of G'. Note also that every cluster $X \in \mathcal{C}'$ has the same number of components in G' as the corresponding cluster $X \setminus \{x'\}$ in G. It remains to show that $(G', \mathcal{C}')$ is c-planar if and only if $(G, \mathcal{C})$ is c-planar.

Clearly, if $(G', \mathcal{C}')$ is c-planar, then so is $(G, \mathcal{C})$, which is an induced clustered subgraph of $(G', \mathcal{C}')$. To prove the converse, assume that we have prescribed cluster regions of $(G, \mathcal{C})$. We may then embed the vertex x' sufficiently close to x, and embed the edges $x'y$ and $x'y'$ sufficiently close to xy and xy', respectively, to ensure that the edge xx' does not intersect any boundary of a cluster region, while the edge $x'y$ intersects the same region boundaries as xy (and similarly for $x'y'$). In this way, we obtain a clustered embedding of $(G', \mathcal{C}')$. $\square$

In the paper we addressed a possible problem with an outer face F of graph G. A pair of vertices of F may still be connected by two non-isomorphic candidate edges belonging to F (see Fig. 1). We also suggested an approach to get rid of this technical complication, by restricting ourselves to properly drawn candidate edges. We now describe this approach formally.

Definition 4. *Let $(G, \mathcal{C})$ be a 2-connected plane clustered graph with no hole. Let $uv \in E(G)$ be an edge on the boundary of the outer face, with the property that the pair of vertices $\{u, v\}$ is not a subset of any non-root cluster (such an edge uv exists, because otherwise the whole outer face would belong to the same* *non-root cluster, forming a hole). Let us fix one such edge uv and call it the* infinite edge*. A candidate edge e is* properly drawn *with respect to the edge uv, if uv is on the boundary of the outer face of $G \cup e$.*

The next lemma shows that to construct a saturator of a 2-connected clustered graph, we may restrict ourselves to properly drawn candidate edges.

The second part of the lemma shows that (under suitable assumptions), we may use an equivalent definition of hole which does not depend on the distinction between inner and outer face of the plane graph G. This will become useful in the description of our algorithm, since it shows that after we restrict ourselves to properly drawn candidate edges, we no longer need to treat the outer face specially.

Lemma 4. *Let $(G, \mathcal{C})$ be a 2-connected plane clustered graph with no hole, let uv be an infinite edge of G.*

1. *If $(G, \mathcal{C})$ has a saturator, then it also has a saturator in which every edge is properly drawn with respect to uv.*
2. *If S is any candidate set containing properly drawn candidate edges, then a cycle c in the graph $G \cup S$ is a hole of a cluster $X \in \mathcal{C}$ if and only if all the vertices of c belong to X and both regions of c contain at least one vertex from $V(G) \setminus X$.*

Proof. Let us start with the first claim of the lemma. Assume that S is a saturator of $(G, \mathcal{C})$, and let G^+ denote the plane graph $G \cup S$. Assume that the edge uv is not on the boundary of the outer face of G^+, otherwise the edges of S are already properly drawn and there is nothing to prove.

Let F be the face of G^+ that contains uv on its boundary and is a subset of the outer face of G. Let A be a point in the interior of F. Let us perform a circular inversion of the plane with center A, followed by a mirror reflection (as shown in Fig. 6). (Strictly speaking, the mirror reflection is not necessary here, however, we apply it anyway, to make the resulting graph 'more similar' to the original one.) Let $\widetilde{G}$, $\widetilde{S}$ and $\widetilde{G}^+$ be the respective images of G, S and G^+ under this transformation. (Less formally, we may say that $\widetilde{G}^+$ is obtained from G^+ by replacing each improperly drawn candidate edge by its properly drawn counterpart, while preserving all the remaining edges.)

Note that the plane graph $\widetilde{G}$ is isomorphic to G, since the center A is in the outer face of G.

Also, all the candidate edges in the set $\widetilde{S}$ are properly drawn with respect to $\widetilde{G}$, because the face F has been transformed into the outer face of $\widetilde{G}^+$. We will now prove that $\widetilde{S}$ is a saturator of $\widetilde{G}$. To see this, it suffices to prove that $\widetilde{G}^+ = \widetilde{G} \cup \widetilde{S}$ has no hole. Assume for contradiction that $\widetilde{G}^+$ has a hole c in a cluster $X \in \mathcal{C}$. Let $w \in V(G) \setminus X$ be a vertex from the inner region of c.

Since no non-root cluster contains both vertices u and v, we may assume that $u \notin X$. Note that u belongs to the outer region of c, because u is on the boundary of the outer face of $\widetilde{G}^+$. It follows that both regions of the cycle c contain a vertex from $V(G) \setminus X$. By the properties of circular inversion, the two

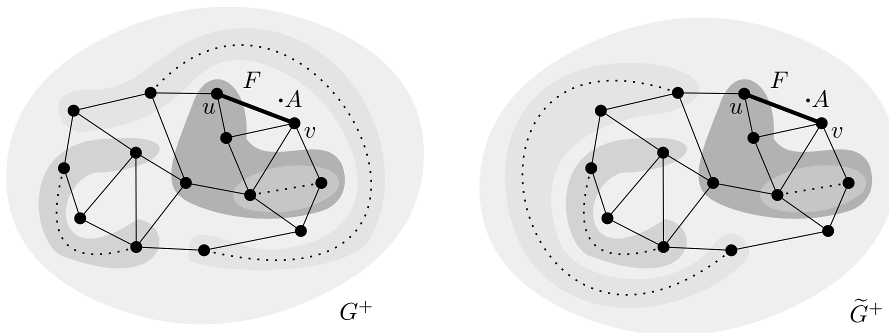

Fig. 6. Illustration to the proof of Lemma 4: applying circular inversion and mirror reflection.

vertices u, w must also belong to different regions of the preimage of c in the graph G^+. We conclude that G^+ also has a hole in the cluster X, which is a contradiction. This proves the first claim of the lemma.

To prove the second claim, we simply observe that if S is a saturator whose edges are properly drawn, then for any cycle c whose vertices all belong to the same non-root cluster X, there must be at least one vertex from $V(G) \setminus X$ in the outer region of c. This is because the two vertices u and v are on the outer face of $G \cup S$, and at least one of them does not belong to X. $\square$

With the following definition and lemma we summarize the information we gained from the previous lemmas.

Definition 5. *A plane clustered graph* $(G, \mathcal{C})$ *is* nice*, if* $(G, \mathcal{C})$ *is 2-component, 2-connected, without holes, with a prescribed infinity edge on the outer face, and with the property that every minimal cluster is disconnected.*

Lemma 5. *If there is a polynomial algorithm that decides whether a nice plane clustered graph has a properly drawn saturator, then there is also a polynomial algorithm deciding c-planarity of general 2-component plane clustered graphs.*

Proof. Assume we want to decide c-planarity of a 2-component plane clustered graph $(G, \mathcal{C})$. Of course, we may assume that $(G, \mathcal{C})$ has no hole, since an existence of a hole can be detected in linear time [3], and a graph without a hole is not c-planar. By Lemma 1, we lose no generality by assuming that every minimal cluster is disconnected. By the Lemmas 3 and 2, we may further assume that G is 2-connected. Finally, the first part of Lemma 4 shows that we may restrict ourselves to searching for properly drawn saturator.

Before we proceed with the description of the general algorithm, we present another lemma, we allows us, under certain assumptions, to simplify the cluster hierarchy. We did not mention this lemma in the main body of our paper, since it is only relevant in situations when the cluster hierarchy is not flat.

Lemma 6. *Let $(G, \mathcal{C})$ be a connected plane clustered graph. Let X be a disconnected cluster, and let $X_1 \subset X$ be a connected component of X. Assume that every subcluster of X that intersects X_1 is a subset of X_1. Let $\mathcal{C}'$ be the set of all the clusters from $\mathcal{C}$ that are subsets of X_1, and let $\mathcal{C}^* = \mathcal{C}' \cup \{V(G), X_1\}$. The following holds:*

1. *If $(G, \mathcal{C}^*)$ is not c-planar, then $(G, \mathcal{C})$ is not c-planar either.*
2. *If $(G, \mathcal{C}^*)$ is c-planar, then any minimal saturator S of $(G, \mathcal{C}^*)$ is harmless for $(G, \mathcal{C})$.*

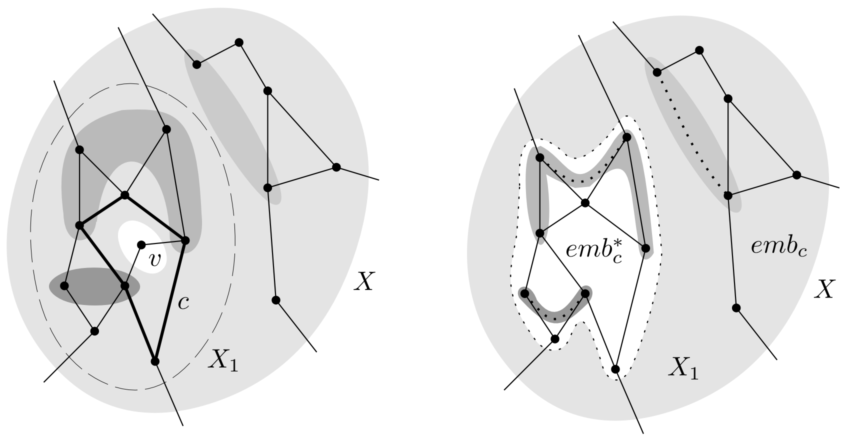

Fig. 7. Illustration to the proof of Lemma 6. Left: a hole c in X_1. Right: combining emb_c and emb_c^* into an embedding of $(G^+, \mathcal{C})$.

Proof. Let us prove the first claim. Assume that $(G, \mathcal{C})$ is c-planar, and let S be its minimal saturator. Let S' be the set of all the candidate edges of S that have both endpoints in X_1. We claim that S' is a saturator of $(G, \mathcal{C}^*)$. It is clear that $G \cup S'$ is a plane graph and that every cluster of $\mathcal{C}^*$ is connected in $G \cup S'$. It only needs to be shown that no cluster of $\mathcal{C}^*$ has a hole in $G \cup S'$. This is only non-obvious for the cluster $X_1 \in \mathcal{C}^*$, since any other cluster in $\mathcal{C}^*$ is also a cluster in $\mathcal{C}$. Assume that X_1 has a hole c in $G \cup S'$ (see Fig. 7). Let v be a vertex from $V(G) \setminus X_1$ in the interior of c. Since the graph G is connected, we may assume that v is connected by an edge of G to a vertex from X_1. This implies that $v \notin X$, because X_1 is a component of X and no vertex from another component of X can be adjacent to a vertex from X_1. It follows that c is also a hole of X in $(G \cup S, \mathcal{C})$, which is impossible. This proves the first claim of the lemma.

To prove the second claim, assume that $(G, \mathcal{C})$ is c-planar (else the claim is trivial). Let us fix, for every cluster $Y \in \mathcal{C}$, a cluster region $emb_c(Y)$. Let $\gamma(Y)$ be the cluster boundary of Y. Fix $\varepsilon > 0$ such that for each cluster $Y \in \mathcal{C}$, every edge of G that does not intersect $\gamma(Y)$ and every vertex of G has distance at least ε from $\gamma(Y)$.

Let us consider a minimal saturator S of $(G, \mathcal{C}^*)$. Let G^+ be the plane graph $G \cup S$. Our aim is to show that S is harmless for $(G, \mathcal{C})$, i.e., $(G^+, \mathcal{C})$ is c-planar. To prove this, we will modify some of the cluster regions $emb_c(Y)$ of those clusters $Y \in \mathcal{C}$ that are subsets of X_1, to obtain a c-planar drawing of $(G^+, \mathcal{C})$, in the way indicated in Fig. 7.

Note that all the vertices of $V(G) \setminus X_1$ are in the outer face of the plane graph $G^+[X_1]$, otherwise $(G^+, \mathcal{C}^*)$ would contain a hole of the cluster X_1 and S would not be a saturator of $(G, \mathcal{C}^*)$. This shows that the edges of S may be embedded in such a way that any point of an edge $e \in S$ has distance at most $\varepsilon/3$ from a point belonging to $G[X_1]$.

Let us now consider the clustered graph $(G^+, \mathcal{C}^*)$. Since each cluster of this graph is connected and has no holes, the graph is c-planar. Let us fix for each cluster $Y \in \mathcal{C}^*$ a cluster region $emb_c^*(Y)$ with boundary $\gamma^*(Y)$. By shrinking the regions if necessary, we may assume that for every cluster $Y \subseteq X_1$, each point of the closed curve $\gamma^*(Y)$ has distance at most $\varepsilon/3$ from the embedding of $G^+[X_1]$, and hence it has distance at most $2\varepsilon/3$ from $G[X_1]$.

Let us now define the cluster regions of the graph $(G^+, \mathcal{C})$. To a cluster $Y \in \mathcal{C}$ that is a subset of X_1, we assign the region $emb_c^*(Y)$, and to a cluster Y that is disjoint from X_1 or a superset of X_1, we assign the region $emb_c(Y)$. It is easy to check that these regions yield a proper c-planar embedding of $(G^+, \mathcal{C})$, showing that S is harmless. $\square$

A.2 The General Algorithm

As in the simple case of flat clustered graphs, we present a polynomial algorithm FIND-EDGE which, for a given nice clustered graph $(G, \mathcal{C})$ as input, will either decide that $(G, \mathcal{C})$ is not c-planar, or it will output a harmless properly drawn candidate edge e. The edge e produced by the algorithm saturates a minimal cluster $X \in \mathcal{C}$, and does not create any hole.

Thus, FIND-EDGE will either show that $(G, \mathcal{C})$ is not c-planar, or it will allow us to reduce the c-planarity of $(G, \mathcal{C})$ to an equivalent nice c-planarity instance $(G \cup e, \mathcal{C}')$, where $\mathcal{C}'$ is obtained from $\mathcal{C}$ by removing all the connected clusters of $G \cup e$ that have no disconnected subclusters. After at most $|C|$ iterations of FIND-EDGE, we will either determine that the original instance $(G, \mathcal{C})$ was not c-planar, or we will obtain a saturator of $(G, \mathcal{C})$.

The algorithm FIND-EDGE will only consider properly drawn candidate edges and it will make no distinction between isomorphic embeddings of the same candidate edge. Thus, for the purpose of this algorithm, a candidate edge is uniquely determined by a pair of its endpoints and the face where the edge should be embedded. After the restriction to properly drawn candidate edges, the algorithm does not need to treat the outer face specially, because, by the second part of Lemma 4, the notion of hole (and hence the notion of saturator) can be defined without referring to the outer face.

In the description of our algorithm, we strive for maximum conceptual simplicity. We thus omit all optimizations that would make the algorithm more efficient but complicate its analysis.

Before we describe the FIND-EDGE algorithm, we need to introduce some terminology. Let $(G, \mathcal{C})$ be a 2-connected 2-component clustered graph. Let $X, Y \in \mathcal{C}$ be a pair of clusters, such that $X \subsetneq Y$. We say that Y is a *connector of* X if X is a subset of a single connected component of Y, otherwise we say that Y is a *disconnector of* X. Note that any disconnector of X is a subcluster of any connector of X. Since every cluster has at least one connector (the root cluster), we may speak of the minimal connector of a cluster X.

Let us now present an outline of the algorithm FIND-EDGE. Assume that $(G, \mathcal{C})$ is a nice clustered graph.

In the first step, the algorithm checks whether there is a non-minimal disconnected cluster $X \in \mathcal{C}$ with components X_1 and X_2, with the property that every subcluster of X is either a subset of X_1 or a subset of X_2. If such a cluster X exists, we may assume w.l.o.g. that X_1 contains at least one subcluster. The algorithm then recursively calls FIND-EDGE on the instance $(G, \mathcal{C}^*)$ described in Lemma 6. If the recursive call determines that $(G, \mathcal{C}^*)$ is not c-planar, then the algorithm outputs that $(G, \mathcal{C})$ is not c-planar and stops. Otherwise, if the recursive call outputs a candidate edge e that is harmless for $(G, \mathcal{C}^*)$, then the algorithm outputs the same edge e and stops. The correctness of this step follows from Lemma 6.

If there is no cluster X that would allow the reduction step described above, we know that every non-minimal disconnected cluster is a disconnector of some minimal cluster. In particular, any candidate set S that saturates every minimal cluster necessarily saturates each cluster of $\mathcal{C}$, because an edge that saturates a minimal cluster X automatically saturates all disconnectors of X as well. This implies that any minimal saturator only contains edges that saturate the minimal clusters.

Let $\mathcal{C}_{\min}$ be the set of all the minimal clusters of $\mathcal{C}$. The clusters $\mathcal{C}_{\min}$ are all disjoint, and by the argument above, any candidate set S that saturates all the clusters of $\mathcal{C}_{\min}$ also saturates the remaining clusters of $\mathcal{C}$. We may be tempted to use the simple FIND-EDGE algorithm to find a minimal saturator S of the flat clustered graph $(G, \mathcal{C}_{\min} \cup \{V(G)\})$, and then use the candidate set S as a saturator of $(G, \mathcal{C})$. The problem with this naive approach is, that the clustered graph $(G \cup S, \mathcal{C})$ may contain a hole in a non-minimal cluster of $X \in \mathcal{C}$. The algorithm we are about to describe is based on the above-described naive approach, but takes additional precautions to avoid problems with holes in non-minimal clusters.

As in the flat case, the algorithm builds a set P of *permitted edges*. In the beginning, the set P contains all the properly drawn candidate edges that saturate a minimal cluster of $\mathcal{C}$.

In the pruning phase, the algorithm repeatedly attempts to remove some candidate edges from P, using a set of *pruning rules*, which we will specify in Subsection A.4.

After none of the pruning rules is applicable, the algorithm performs a series of triviality checks, which are analogous to those used in the flat case:

1. If there is a minimal cluster that cannot be saturated by any permitted edge, the algorithm declares that $(G, \mathcal{C})$ is not c-planar and stops.
2. If there is a minimal cluster that can be saturated by exactly one permitted edge $e \in P$, the algorithm outputs e as a harmless candidate edge and stops.
3. The algorithm looks for a permitted edge e which does not 'obstruct' any other permitted edge (in a sense that will be made precise later). If such an edge exists, the algorithm outputs e as a harmless edge and stops.

The triviality checks will be described in greater detail in Subsection A.5, when we will also prove their correctness.

If the triviality checks fail, the algorithm will proceed to the last phase, which is again analogous to the flat case. In this phase, the algorithm will attempt to locate a harmless edge among the permitted edges. Depending on the structure of P, the algorithm will use one of two possible procedures:

– If there is a minimal cluster X whose permitted saturating edges all belong to the same face F of G, then the algorithm will use a procedure LOCATE-IN-FACE, to find a harmless permitted edge that belongs to the face F.
– If each minimal cluster can be saturated in at least two distinct faces, it can be shown that every cluster can in fact be saturated in exactly two faces. Furthermore, it can be shown that in such situation, any permitted edge of P is harmless. Thus, the algorithm will perform a procedure called OUTPUT-ANYTHING, which outputs an arbitrary permitted edge and stops.

The procedures LOCATE-IN-FACE and OUTPUT-ANYTHING will be described and analyzed in Subsections A.6 and A.7, respectively. Each of these procedures is guaranteed to output the correct solution.

A.3 Properties of 2-Component Graphs

Before we describe in detail the building blocks of the algorithm outlined above, we will establish some basic properties of 2-component clustered graphs, and introduce some more terminology.

Let G be a 2-connected plane graph. Let a, b, c, d be a quadruple of distinct vertices on the boundary of a face F of G. Recall that the pair *ab crosses* the pair *cd* in F, if the four vertices appear on the boundary of F in the cyclic order *acbd*. If e and f are two candidate edges of a 2-connected clustered graph $(G, \mathcal{C})$, we say that e *crosses* f if the two candidate edges belong to the same face F of G and the endpoints of e cross with the endpoints of f. For two sets of vertices X and Y, we say that X *crosses* Y *in face* F, if there are vertices $a, b \in X$ and $c, d \in Y$ such that a, b crosses c, d in the face F.

We first prove two simple lemmas which describe general properties of connected subgraphs in plane graphs.

Lemma 7. *Let G be a 2-connected plane graph. If X and Y are disjoint sets of vertices which both induce a connected subgraph of G, then X does not cross Y in any face of G.*

Proof. Assume for contradiction that there are four vertices x_1, x_2, y_1, y_2 on the boundary of a face F, with $x_i \in X$ and $y_i \in Y$, and that the pair x_1x_2 crosses y_1y_2 in F. Let p be a path in $G[X]$ connecting the vertices x_1 and x_2. Let us draw a new edge e with endpoints x_1 and x_2 in the interior of F (if x_1 and x_2 are already connected by an edge of G, we may subdivide e by a new vertex to avoid multiple edges in $G \cup e$). In the graph $G \cup e$, there is a cycle $p \cup e$. The vertices y_1 and y_2 belong to different regions of this cycle, because they can be connected by a curve drawn inside F which intersects the cycle exactly once. This is a contradiction, because y_1 and y_2 are connected by a path in $G[Y]$ which does not intersect $p \cup e$. □

Lemma 8. *Let G be a 2-connected plane graph. Let X_1, X_2, and X_3 be disjoint sets of vertices, each of them inducing a connected subgraph of G. Then there are at most two distinct faces of G that contain vertices of all the three sets X_i on their boundaries.*

Proof. Assume that there are three faces F_1, F_2 and F_3, and each of them has at least one vertex from each of the three sets X_1, X_2, and X_3. Place a new vertex into the interior of each of the three faces, and connect each of the three new vertices by an edge to the vertices of X_1, X_2, and X_3 on the boundary of the respective faces. The resulting graph is planar but contains a $K_{3,3}$ minor, which is impossible. □

Throughout the rest of this subsection, we assume that $(G, \mathcal{C})$ is a nice clustered graph. We also assume that all the candidate edges are properly drawn.

Lemma 9. *Let S be a candidate set of $(G, \mathcal{C})$, such that $(G \cup S, C)$ has no hole. Let e be a candidate edge of the graph $G \cup S$ such that $(G \cup S \cup \{e\}, \mathcal{C})$ contains a hole belonging to a cluster $X \in \mathcal{C}$. Then one of the following two possibilities holds:*

1. *Both endpoints of e belong to the same component of $G[X]$ and X has a hole in $(G \cup e, \mathcal{C})$.*
2. *The two endpoints of e belong to distinct components of $G[X]$, and there is a candidate edge $f \in S$ with both endpoints belonging to distinct components of $G[X]$ and such that X has a hole in $(G \cup \{e, f\}, \mathcal{C})$.*

Proof. Clearly, both endpoints of e must belong to the cluster X. Assume that the two endpoints of e belong to the same component X_1 of $G[X]$ (as in the left part of Fig. 8). Let G_X be the plane subgraph of $G \cup S \cup \{e\}$ induced by the vertices of X. By assumption, G_X has a cycle c which is a hole in $G \cup S \cup \{e\}$. Hence, both regions of c in $G \cup S \cup \{e\}$ contain a vertex of G not belonging to X (here we use the alternative definition of hole deduced from the second part of

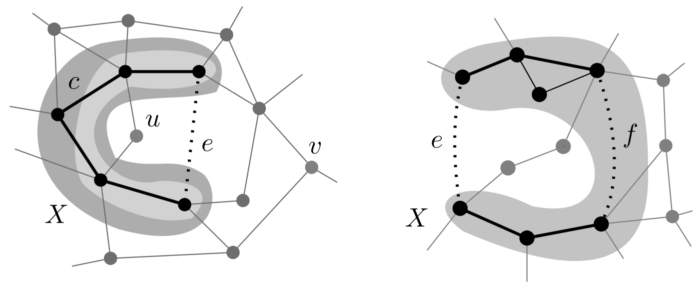

Fig. 8. Two possibilities of a hole caused by a candidate edge e.

Lemma 4). Let $u, v \in \overline{X}$ be two vertices in distinct regions of c. Since no cycle of $G_X - e$ is a hole in $G \cup S \cup \{e\}$, we know that the two vertices u, v are in the interior of the same face of $G_X - e$. Hence, the two vertices are in the interior of the two faces of G_X that are adjacent to e.

Let p be a path in $G[X]$ that connects the two endpoints of e (recall that these endpoints belong to the same component of $G[X]$). Then $p \cup e$ is a cycle in G_X, and each region of this cycle contains one face of G_X adjacent to e. In particular, $p \cup e$ is a hole in $G \cup e$.

Assume now that the endpoints of e belong to distinct components of $G[X]$ (see the right part of Fig. 8). Since in $G \cup S \cup e$, the cluster X contains a cycle (in fact a hole) containing the edge e, the set S must contain at least one edge that saturates X. Let f be such an edge. Then X induces a connected subgraph in the graph $G \cup f$. We may repeat the argument of the previous paragraph for the graph $G \cup f$ instead of G, and the set $S \setminus \{f\}$ instead of S, to conclude that X has a hole in $(G \cup \{f, e\}, \mathcal{C})$. □

Definition 6. *Assume that $(G, \mathcal{C})$ is a nice clustered graph. We say that a candidate edge e is* blocked *by a cluster $X \in \mathcal{C}$ if X has a hole in $(G \cup e, \mathcal{C})$. We say that a candidate edge e* interferes with a candidate edge f *in a cluster X if both e and f saturate X, and X has a hole in $(G \cup \{e, f\}, \mathcal{C})$.*

Lemma 10. *A set S of properly drawn candidate edges is a partial saturator of $(G, \mathcal{C})$ if and only if no edge of S is blocked, no two edges of S interfere, and no two edges of S cross.*

Proof. If S is a partial saturator, it satisfies the three conditions by definition. Assume now that S is not a partial saturator. If two edges of S cross, we are done, so assume that no two edges cross. It follows that $G \cup S$ has a hole. Let S_0 be a minimal subset of S such that $G \cup S_0$ has a hole. From Lemma 9 we see that every edge of S_0 is either blocked or interferes with another edge of S_0, as claimed. □

Let F be a face of G, let W be a set of vertices. A W*-arc of* F is an arc of the curve forming the boundary of F which has two vertices of W as endpoints and contains no vertices of W in its interior. If the boundary of F has less than two vertices from W, we assume that the whole boundary forms a single W-arc. Clearly, if the boundary of F has $k \geq 1$ vertices of W, then F has exactly k internally disjoint W-arcs.

Note that if two adjacent faces F and F' share a common part of their boundary, then a W-arc of F may correspond to the same curve as a W-arc of F'. Nevertheless, we will treat the two arcs as two distinct objects, even if they are represented by the same curve.

Definition 7. *Let X be a disconnected cluster with components X_1 and X_2. Let F be a face whose boundary contains at least one vertex from each component of X. Since X_1 and X_2 cannot cross in F by Lemma 7, we see that among all the X-arcs of F there are exactly two arcs whose one endpoint belongs to X_1 and the other to X_2. We will call these two arcs* the external X-arcs of F. *The remaining X-arcs will be called* internal. *More generally, if X' is a subset of X, such that both of the sets $X' \cap X_1$ and $X' \cap X_2$ have at least one vertex on the boundary of F, then an X'-arc of F is called an* external X'-arc *if it has one endpoint in X_1 and another endpoint in X_2.*

Lemma 11. *Let X and Y be disjoint disconnected clusters, with components X_1, X_2 and Y_1, Y_2, respectively. Let F be a face whose boundary contains vertices of all the four components. The following holds:*

1. *If a vertex of Y_1 appears in an internal X-arc of F, then all the vertices of Y_1 on the boundary of F appear in the same X-arc.*
2. *If the vertices of Y appear in more than one X-arc of F, and at least one vertex of Y appears in an internal X-arc, then F is the only face of G that contains vertices from both of the sets Y_1 and Y_2.*

Proof. Let us prove the first claim. Assume that a vertex $y \in Y_1$ belongs to an internal X-arc, whose endpoints belong to the same component X_1 of the cluster X. If there is another X-arc that contains a vertex from Y_1, then the two sets X_1 and Y_1 cross in F, which contradicts Lemma 7.

To prove the second claim, assume again that a vertex $y_1 \in Y_1$ belongs to an internal X-arc with endpoints in X_1, and that there is a vertex $y_2 \in Y_2$ that belongs to another X-arc. If there is another face F' which contains a vertex $y_1' \in Y_1$ as well as a vertex $y_2' \in Y_2$, then let us draw a new edge $y_1'y_2'$ in the interior of the face F'. In the resulting graph, the set Y induces a connected subgraph, and yet Y crosses X_1 in the face F, which contradicts Lemma 7. □

We will now use the concept of arcs to characterize blocking and interference of candidate edges.

Lemma 12.

1. *Let $X \in \mathcal{C}$ be a cluster, let e be a candidate edge in a face F. The edge e is blocked by X if and only if both endpoints of e belong to the same component of X and the two endpoints appear in distinct $\overline{X}$-arcs of F.*

2. *If a candidate edge e that saturates a disconnected cluster Y is blocked by a cluster X, then e is also blocked by the minimal connector of Y (which is a subcluster of X).*

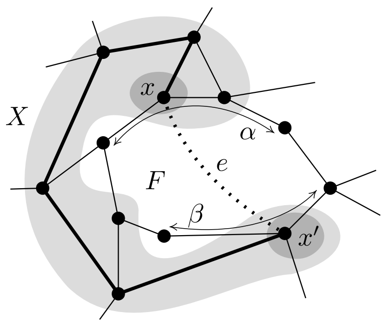

Fig. 9. Illustration to Lemma 12: x and x' in different $\overline{X}$-arcs α and β of F.

Proof. See Fig. 9. Let us begin with the first part of the lemma. Let x and x' be the endpoints of e. Assume first that e is blocked by X. This implies that both endpoints of e belong to the same component of X. For contradiction, suppose that x and x' also belong to the same $\overline{X}$-arc of F. In particular, x and x' split the boundary of F into two arcs α and β, and at least one of these arcs (let us say the arc α) only contains vertices from X.

Let G_X be the subgraph of the plane graph $G \cup \{e\}$ induced by the vertices of X. By assumption, the graph G_X has a cycle c such that both regions of c have at least one vertex of $\overline{X}$ in their interior. As in Lemma 9, we conclude that there are vertices $u, v \in \overline{X}$ that belong to the interiors of the two faces of G_X that are adjacent to e. However, this is impossible, since the arc α together with the edge e forms the boundary of a face of G_X that has no other vertex of G in its interior. We conclude that if e is blocked by X, then x and x' must belong to different $\overline{X}$-arcs.

To prove the converse, assume that x and x' belong to different $\overline{X}$-arcs of F and that they belong to the same component of X. Let α and β be again the two arcs of F determined by the pair of vertices x and x'. Since x and x' are in distinct $\overline{X}$-arcs of F, both α and β must contain a vertex from $\overline{X}$. Let us choose $u \in \alpha \cap \overline{X}$ and $v \in \beta \cap \overline{X}$. By assumption, there is a path p in G from x to x' which only contains the vertices of X. We see that $p \cup e$ is a cycle in $G \cup e$ and that u and v belong to different regions of this cycle. It follows that $p \cup e$ is a hole of X, showing that e is blocked by X, as claimed.

To prove the second claim of the lemma, assume that Z is the minimal connector of Y. Since X is also a connector of Y, we have $Z \subseteq X$. It follows that a pair of vertices that belongs to different $\overline{X}$-arcs of a face F must also belong to different $\overline{Z}$-arcs of F. $\square$

The next lemma provides a similar characterization for edge-interference.

Lemma 13.

1. *Let $e = ab, f = cd$ be two non-crossing candidate edges. Let Z be a disconnected cluster with components Z_1 and Z_2, and assume that $\{a, c\} \subseteq Z_1$ and $\{b, d\} \subseteq Z_2$. Then e and f do not interfere in Z if and only if e and f belong to the same face F of G, a and c belong to the same $\overline{Z}$-arc of F, and b and d belong to the same $\overline{Z}$-arc of F.*
2. *If e and f saturate two distinct minimal clusters X and Y, then e and f interfere if and only if they interfere in the minimal common disconnector of the two clusters X and Y.*

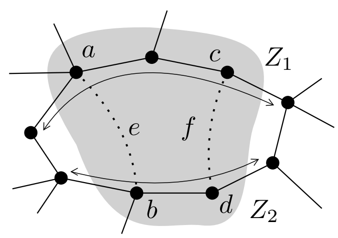

Fig. 10. Illustration to Lemma 13: two non-interfering candidate edges.

Proof. See Fig. 10. Note that if Z_1 and Z_2 are two components of a disconnected cluster Z, then a vertex of Z_1 always belongs to a different $\overline{Z}$-arc than any vertex of Z_2.

The first part of Lemma 13 follows from Lemma 12, using the fact that e interferes with f in a cluster Z if and only if e is blocked by Z in the graph $G \cup f$. The second part of the lemma is analogous to the second part of Lemma 12. $\square$

A.4 Pruning Phase

Let us assume that $(G, \mathcal{C})$ is a nice clustered graph. We also assume that every disconnected cluster that is not minimal is a disconnector of a minimal cluster. Recall that a nice clustered graph that does not satisfy this last assumption can be reduced using Lemma 6 to an equivalent instance of c-planarity with the same underlying graph and fewer disconnected clusters.

In the pruning phase, the algorithm FIND-EDGE iteratively restricts the set P of permitted candidate edges. In the beginning of the pruning phase, the set P is initialized to contain all the properly drawn candidate edges that saturate at least one minimal cluster. Note that every permitted edge $e \in P$ saturates a unique minimal cluster $X \in \mathcal{C}$. A permitted edge that saturates a minimal cluster X will be called an *X-edge*.

We say that a cluster X *interferes* with a cluster Y, if P contains an X-edge e and a Y-edge f that interfere in some cluster Z. Note that in such case, the two edges e and f must interfere in the minimal common disconnector of X and Y (this follows from Lemma 13). Note also, that since edges are removed from P in the pruning phase, two clusters that interfere at the beginning of the pruning do not necessarily interfere when the pruning is over.

Before we describe the pruning rules, let us introduce more terminology.

Let F be a face, and let X be a minimal cluster. We say that a vertex v is *active* in F (or F-*active*), if it is incident to at least one permitted edge that belongs to F, otherwise it is *passive*. Note that the same vertex may be passive in one face and active in another.

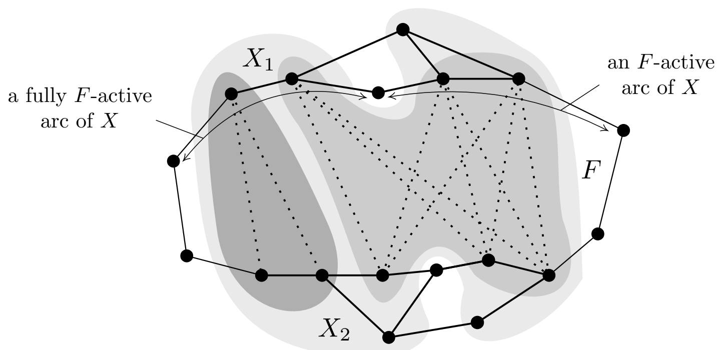

Fig. 11. Examples of active and fully active arcs of a cluster X in a face F.

Let X be a disconnected non-minimal cluster, let F be a face. We say that an $\overline{X}$-arc α of the face F is an F-*active arc of cluster* X if α contains at least one F-active vertex that belongs to a cluster disconnected by X (or equivalently, if there is a permitted edge in F that saturates X and has an endpoint in α). We say that α is a *fully* F-*active arc of* X if every minimal cluster that is disconnected by X has at least one active vertex in α (for illustration, see Fig. 11). If X_1 and X_2 are the two components of X, then any $\overline{X}$-arc may only contain vertices of at most one of the two components. An F-active arc of X that contains vertices from X_1 will be simply called an F-*active arc of* X_1. If the face F is clear from the context, we will write 'active' instead of 'F-active'.

If X is a minimal cluster, and if e and e' are two X-edges, we say that e and e' are *equivalent*, if for every permitted edge $f \in P$ that is not an X-edge, the edge f crosses e if and only if it crosses e', and f interferes with e if and only if it interferes with e'.

During the pruning phase, the set P will satisfy the following invariants.

(I1) For every minimal cluster X and every face F, all the X-edges that belong to F form a vertex-disjoint union of complete bipartite subgraphs; these complete bipartite subgraphs will be called X*-bundles* (or just *bundles*, if X is clear from the context). Two X-edges belonging to different bundles do not cross.

(I2) If X and Y are distinct minimal clusters, then if an X-edge e crosses two Y-edges f and f', then f and f' belong to the same bundle.

(I3) Let X be a disconnected non-minimal cluster, let $X' \subset X$ be a component of X. If Y is a minimal cluster disconnected by X, and if, in a face F, the component X' has more than one F-active arc containing active vertices of Y, then no F-active arc of X' may contain vertices from two distinct bundles of Y (see Fig. 12).

(I4) If $(G, \mathcal{C})$ is c-planar, then it has a saturator that is a subset of P.

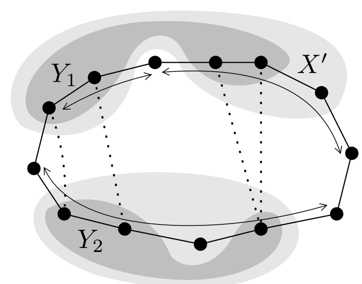

Fig. 12. An example of a configuration forbidden by I3.

These are the four rules applied to reduce P during the pruning phase.

(R1) If there is a disconnected non-minimal cluster X that has, in a face F, an active arc α that is not fully active, then remove from P all the edges from the face F that saturate X and have an endpoint in α. Thus, the arc α will no longer be an F-active arc of X.

(R2) If, in a face F, there is a permitted edge which is blocked by a cluster Y, then remove from P all the permitted edges in F blocked by Y. (See Fig. 9 for an example of a blocked edge.)

(R3) If, for a minimal cluster X, there is a permitted edge that crosses all the X-edges, then remove from P each edge that crosses all the X-edges.

(R4) Let $e = uv$ and $e' = u'v$ be two X-edges that belong to the same face F and that share a common vertex v. If e and e' are equivalent then remove from P all the X-edges in F incident to u'.

The four rules above are listed in order of decreasing precedence. In the pruning phase, the FIND-EDGE algorithm tests the rules, starting from the first one, to find the first rule that is applicable for the given set P. After a rule is applied, the algorithm returns to the beginning of the list of rules, and starts looking for the first rule that is applicable for the (now smaller) set P. In particular, the algorithm only applies rule number i in a situation when none of the rules $1, \ldots, i-1$ is applicable. Note that if for a given set P the second rule is not applicable (i.e., none of the permitted edges is blocked), then it will never become applicable if the set P is further reduced, so it will never be used again.

Our main task is to show that all the invariants are preserved during the pruning phase. Let us first verify that before any of the rules is applied, the initial set P satisfies the invariants. The validity of I1 is clear, since initially, for every face F and every minimal cluster X, all the properly drawn candidate edges in F that saturate X belong to P, in particular, there is at most one X-bundle in every face. This also shows that I2 and I3 hold trivially (there is only one X-bundle in each face). The validity of I4 is also clear, since if $(G, \mathcal{C})$ is c-planar, then it has a saturator whose every edge is properly drawn (Lemma 4, part 1); furthermore, every disconnected cluster is a disconnector of a minimal cluster, so if S is a minimal properly drawn saturator, then $S \subseteq P$.

We will prove that the four invariants remain valid when P is pruned by the four rules above. More precisely, we will show that if a rule is applied to a set P which satisfies the invariants and which cannot be pruned by any of the higher-precedence rules, then after the application, the remaining permitted edges will still satisfy the invariants. We will deal with the four rules separately in Lemmas 14, 16, 17 and 19. In the proofs of these lemmas, we let P_{B} denote the set of permitted edges *before* the application of the rule, and we let P_{A} denote the set of permitted edges *after* the application of the rule. In each of the four proofs, we assume that P_{B} satisfies all the invariants and that it cannot be pruned by any higher-precedence rule; we then show that P_{A} also satisfies the invariants.

Lemma 14. *The rule R1 preserves all the invariants.*

Proof. The rule R1 merely reduces the size of existing bundles by removing some of their vertices. In particular, if a pair of edges in P_{A} belongs to two distinct bundles, then it also belongs to two distinct bundles of P_{B}. Thus, R1 clearly preserves the first three invariants.

We claim that it preserves the last invariant. To see this, note that for any saturator $S \subseteq P_{\mathrm{B}}$ and any disconnected non-minimal cluster X, all the edges of S that saturate the cluster X must belong to the same face, and their endpoints must belong to a union of two $\overline{X}$-arcs of this face, otherwise S would contain a pair of interfering edges due to Lemma 13. This implies that all the edges of S that saturate X must have their endpoints in fully active arcs of S.

Since R1 removes edges that saturate X but have an endpoint in an arc that is not fully active, we know that $S \subseteq P_{\mathrm{A}}$. Thus R1 preserves I4. □

Before we deal with the remaining rules, we need a lemma that describes the situations when the rule R1 is not applicable.

Lemma 15. *Assume that the set P of permitted edges cannot be pruned by the rule R1. Let Z be a disconnected cluster that is a disconnector of at least two distinct minimal clusters X and Y. Let Z_1 and Z_2 be the two components of Z. The following holds:*

1. *All the active arcs of Z are fully active.*
2. *There are at most two faces that have active arcs of Z.*
3. *If there are exactly two distinct faces F and F' that have active arcs of Z, then each of these two faces has exactly one active arc of Z_1 and exactly one active arc of Z_2.*
4. *If there is only one face F that has active arcs of Z, then F has at most two active arcs of Z_1 and at most two active arcs of Z_2.*

Proof. Let X_1 and X_2 be the components of X, let Y_1 and Y_2 be the components of Y, and assume that $X_i \cup Y_i \subseteq Z_i$ for $i = 1, 2$. The first claim of the lemma follows from the fact that R1 is not applicable in P.

To prove the second claim, note that if there were three faces with active arcs, then by the first part, all the three faces would contain vertices from all the four disjoint sets X_1, Y_1, X_2 and Y_2, contradicting Lemma 8.

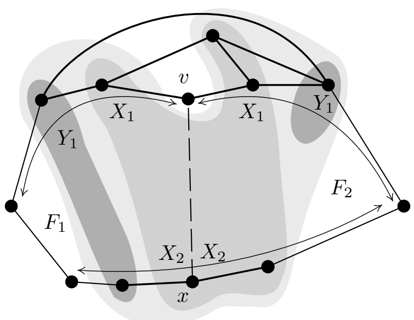

Fig. 13. Illustration to the proof of Lemma 15: too many active arcs in a face.

To prove the third claim, assume for contradiction that there are two distinct faces F and F' with active arcs of Z, and that the face F has at least two active arcs of Z_1. Let x be a vertex from X_2 that belongs to the boundary of F. Choose a vertex v on the boundary of F in such a way that the two vertices x and v separate the boundary of F into two arcs, and each of these two arcs contains one active arc of Z_1. Draw a new edge e with endpoints x and v into the face F (see Fig. 13). This subdivides F into two new faces F_1 and F_2, each of them containing at least one vertex from each of the three sets X_1, Y_1 and X_2. Since the face F' is also intersected by these three sets, the graph $G \cup e$ contradicts Lemma 8.

The last claim of the lemma is proved by an analogous argument. Assume that F is a face that contains three active arcs of Z_1. By inserting two new edges into F, we may subdivide it into three faces, each of them intersected by each of the three sets X_1, Y_1 and X_2. This contradicts Lemma 8. $\square$

Lemma 16. *The rule R2 preserves all the invariants.*

Proof. Let Y and F be as in the description of the rule R2, let X be a minimal cluster whose candidate edge has been pruned by the current application of R2. By Lemma 12, the rule R2 has removed from P_{B} precisely those permitted edges in F whose endpoints belong to Y, but do not belong to the same $\overline{Y}$-arc of F. Let X_1 and X_2 be the two components of X. Since X_1 may not cross X_2, there can be at most two $\overline{Y}$-arcs α and β that contain vertices from both X_1 and X_2. Thus, a given X-bundle of P_{B} in the face F with vertex set $W \subseteq X$ will be transformed into two (possibly empty) bundles of P_{A}, induced respectively by the vertices of $W \cap \alpha$ and $W \cap \beta$. This shows that I1 is preserved.

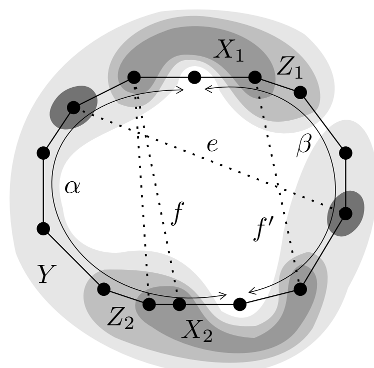

Fig. 14. Illustration to the proof of Lemma 16: an edge e crossing two bundles.

Next, consider the invariant I2: assume, for contradiction, that an edge $e \in P_{\mathrm{A}}$ crosses a pair of permitted X-edges f and f' that belong to distinct bundles of P_{A} (see Fig. 14). This is only possible if f has both endpoints in the arc α and f' has both endpoints in the arc β (or vice versa). It follows that e must have one endpoint in the interior of α and the other in the interior of β, hence e is blocked by Y and should have been removed by the current application of R2.

Let us assume now that P_{A} violates I3. This is only possible if, for a minimal cluster X whose permitted edges have been pruned by the current application of R2, there is a disconnector Z of X such that an F-active arc of Z contains two F-active vertices u, v of X from distinct bundles of P_{A}. Since I3 holds in P_{B}, we know that u, v belong to the same bundle of P_{B} and hence $u \in \alpha$ and $v \in \beta$. However, every disconnector of X is a subset of any connector of X, in particular $Z \subset Y$, and since u, v are in distinct $\overline{Y}$-arcs, they cannot be in the same active arc of Z.

It is clear that R2 preserves I4, since a saturator $S \subseteq P_{\mathrm{B}}$ may not contain an edge that is blocked by some cluster, and hence $S \subseteq P_{\mathrm{A}}$. □

Lemma 17. *The rule R3 preserves all invariants.*

Proof. Let X be a minimal cluster, such that there is at least one edge in P_{B} that crosses all the X-edges. It follows that all the X-edges of P_{B} belong to the same face F. Let X_1 and X_2 be the two components of X, let X^a be the set of F-active vertices of F. Let α and β be the two external X^a-arcs of F (the notion of external arcs has been introduced in Definition 7).

Let us argue that R3 preserves I1. Let Y be a minimal cluster whose permitted edges have been pruned by R3. Let $e \in P_{\mathrm{B}} \setminus P_{\mathrm{A}}$ be a Y-edge that has been pruned. Since e intersects all the X-edges, e must have an endpoint in α and another endpoint in β. From Lemma 11, we obtain that all the vertices of Y on the boundary of F belong to the external X^a-arcs. Note that an edge of P_{B} is pruned by this application of R3 if and only if one of its endpoints belongs to α and the other belongs to β. In particular, a bundle of Y with vertex set W is split into (at most) two bundles, induced by the vertex sets $W \cap \alpha$ and $W \cap \beta$. Thus, I1 is preserved.

Assume that P_{A} violates I2. Then there is an edge $e \in P_{\mathrm{A}}$ which intersects a pair of Y-edges f and f' that belong to the same Y-bundle in P_{B}, but they belong to distinct bundles in P_{A}. This implies that e has an endpoint in α and another endpoint in β, and should have been removed from P_{A} as well. Let us

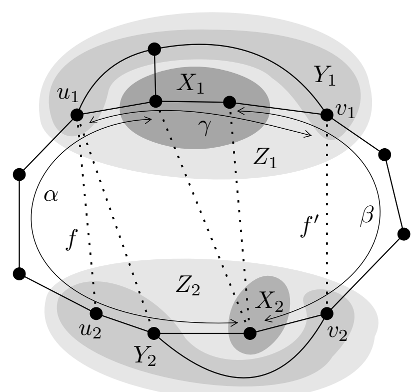

Fig. 15. Illustration to the proof of Lemma 17: edges in bundles of Y and active arcs.

argue that R3 preserves I3 (see Fig. 15). Clearly, P_{A} may only violate I3 if P_{A} contains a pair of Y-edges f, f' that belong to the same bundle in P_{B} and to distinct bundles in P_{A}. Furthermore, for some disconnector Z of Y, there must be an active arc γ of Z in the face F that contains an endpoint of f as well as an endpoint of f'. Our aim is to show that under these assumptions, the component of Z that has the active arc γ may not have any other active arc in F, which will imply that the invariant I3 is preserved.

To see this, let $f = u_1u_2$, let $f' = v_1v_2$, let Z_1, Z_2 be the components of Z, and assume that $\{u_i, v_i\} \subseteq Y_i \subseteq Z_i$ for $i = 1, 2$. Since Y_1 cannot cross with Y_2, and since X cannot cross with f or f', we may assume w.l.o.g. that the vertices of X_1 on the boundary of F belong to an internal Y-arc whose endpoints are in Y_1, and similarly the vertices of X_2 belong to an internal Y-arc with endpoints in Y_2. W.l.o.g. we may assume that the active arc γ belongs to Z_1, hence it contains both u_1 and v_1. Then γ contains all the vertices of X_1 in F, hence $X_1 \subseteq Z_1$. Note that X_2 cannot be a subset of Z_1, because Z_1 would cross Z_2. We conclude that $X_2 \subset Z_2$ and Z is a disconnector of X. Since every active arc must be fully active by Lemma 15, and all the vertices of X_1 on the boundary of F belong to the active arc γ, we see that γ is the only active arc of Z_1 in F, and I3 is preserved.

It is clear that R3 preserves I4, since a saturator $S \subseteq P_{\mathrm{B}}$ cannot contain an edge that crosses all the permitted edges saturating X. $\square$

Before we deal with the last rule, we prove the following lemma.

Lemma 18. *Let P be a set of permitted edges in which the rule R1 is not applicable. Let X be a minimal cluster with components X_1 and X_2. Let $u, u' \in X_1$ be a pair of vertices belonging to the same X-bundle of P, let $v \in X_2$ be another vertex from the same bundle. Let F be the face that contains the bundle, let α be the arc of F whose endpoints are the vertices u, u' and which does not contain v. The two permitted edges uv and $u'v$ are not equivalent if and only if at least one of the following two conditions hold (see Fig. 16):*

- *There is an edge $f \in P$ that is not an X-edge and has exactly one endpoint in α.*
- *There are clusters Y and Z, such that Z is a disconnector of X and Y, and the arc α contains at least one vertex from $\overline{Z}$.*

In particular, if $v' \in X_2$ is a vertex that belongs to the same bundle as the vertices u, u' and v, then the edges uv and $u'v$ are equivalent if and only if the edges uv' and $u'v'$ are equivalent.

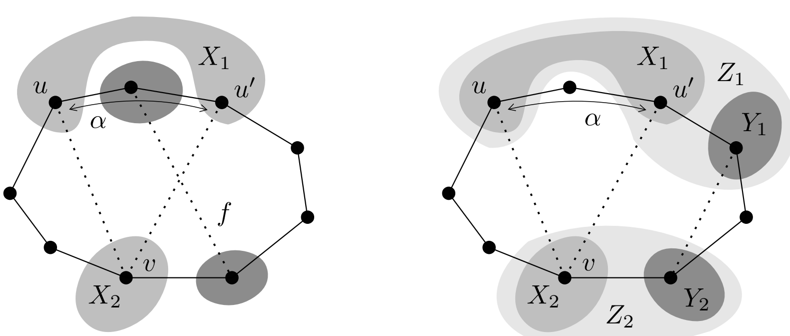

Fig. 16. Two candidate edges uv and $u'v$ that are non-equivalent because of a crossing (left), and because of interference (right).

Proof. Let us show that each of the two conditions implies that uv and $u'v$ are not equivalent. For the first condition, this is obvious.

Let us assume that the second condition holds. Let Z_1 be the component of Z that contains u and u'. By the second condition, u and u' belong to distinct active arcs of Z_1. Let y be an F-active vertex of Y that belongs to the same active arc as v, and let f be a Y-edge incident to y and belonging to F. The edge f must have an endpoint either in the same active arc as u or in the same active arc as u', since Z_1 cannot have more than two active arcs in F. In any case, f interferes with exactly one of the two X-edges uv and $u'v$, showing that these two edges are not equivalent.

Assume now that the two edges uv and $u'v$ are not equivalent. Then there is an edge $f \in P$ that is not an X-edge, and which either crosses or interferes with exactly one of the two edges uv and $u'v$. If f crosses exactly one of the two edges, then the first condition holds. If f interferes with exactly one of the two edges, then the second condition holds.

The last claim of the lemma follows from the fact that the two conditions of the first part are independent of the choice of v. □

Lemma 19. *The rule R4 preserves all invariants.*

Proof. The rule R4 reduces the sizes of the bundles by removing active vertices. However, it does not split a bundle into two (or more) nonempty bundles. Thus, it clearly preserves all the first three invariants.

Let us argue that it also preserves I4. Fix u, u', v and X as in the description of the rule R4. In particular, uv and $u'v$ are equivalent X-edges and all the X-edges of the face F incident to u' have been pruned. Let $S' \subseteq P_{\mathrm{B}}$ be a minimal saturator. Then S' has exactly one X-edge e'. If e' is not incident to u', we are done because $S' \subseteq P_{\mathrm{A}}$. Otherwise, let $e' = u'v'$, where v' is a vertex of the same bundle as u, u' and v. By Lemma 18, e' is equivalent to the edge $e = uv'$. From Lemma 10 it follows that the set $S = S' \cup \{e\} \setminus \{e'\} \subseteq P_{\mathrm{A}}$ is a saturator. □

This completes the description of the pruning phase and the proof of its correctness.

A.5 Triviality Checks

When there is no rule applicable to the set P of permitted edges, the pruning phase ends. The FIND-EDGE then proceeds with a series of three types of triviality checks, described below.

First, the algorithm checks whether there is a minimal cluster X that is not saturated by any permitted edge. If this is the case, the algorithm FIND-EDGE concludes that the clustered graph $(G, \mathcal{C})$ is not c-planar and stops. This is a correct conclusion, since if $(G, \mathcal{C})$ were c-planar, then by the invariant I4 there would have to be a saturator made of permitted edges, which is clearly impossible.

As the next triviality check, the algorithm tries to find a minimal cluster X, such that the set P contains a single X-edge e. If such a cluster X is found, the algorithm outputs e as a harmless edge and stops. This is again a correct output, since by I4, if G is c-planar, then it has a saturator S which is a subset of P. Necessarily, S contains the edge e. This implies that e is harmless.

In the last type of triviality check, the algorithm looks for a permitted edge e that neither crosses nor interferes with any permitted edge belonging to a different minimal cluster. If such an edge e is found, the algorithm outputs e as a harmless edge and stops. To see that this output is correct, assume that $(G, \mathcal{C})$ is c-planar, and let X be the minimal cluster saturated by e. Let $S' \subseteq P$ be a minimal saturator. S' contains exactly one X-edge e'. The set $S = (S' \setminus \{e'\}) \cup \{e\}$ has no pair of crossing or interfering edges and it saturates the same minimal clusters as S'. Hence, S is a saturator, which shows that e is harmless.

If none of the triviality checks succeeds, the algorithm counts, for each minimal cluster X, the number of faces of G that contain at least one X-edge. We will say that a minimal cluster X is *one-faced* if all the X-edges belong to a single face of G, X is *two-faced* if all the X-edges appear in the union of two distinct faces, and X is *many-faced* otherwise.

Recall that if there is a one-faced cluster X, whose permitted edges belong to a face F, then the algorithm performs a subroutine LOCATE-IN-FACE to find a harmless permitted edge in F. This subroutine is described in Subsection A.6.

If there is no one-faced cluster, we will later prove that all the minimal clusters are two-faced, and that any edge is harmless. The algorithm then outputs an arbitrary permitted edge and stops. This step is described in detail in Subsection A.7.

A.6 LOCATE-IN-FACE

Assume that we are given a set P of permitted edges satisfying all the invariants I1–I4 described in Subsection A.4. Assume furthermore that P cannot be pruned by any of the rules R1–R4, and that none of the triviality checks is applicable.

For a face F, we shall say that a minimal cluster X is an *F-cluster*, if all the X-edges belong to F.

Assume that we are given a face F with at least one F-cluster. Before we describe the procedure LOCATE-IN-FACE, we need some preparation.

Lemma 20. *Let X be an F-cluster with components X_1 and X_2. Let Y be another minimal cluster. Assume that there is a Y-edge f that crosses at least one X-edge e. The following holds:*

1. *Y is an F-cluster.*
2. *One of the components of Y has all its active vertices in a single internal X-arc α of F. The other component of Y has at least one F-active vertex in each of the two external X-arcs.*
3. *The X-edges form at most two bundles.*

Proof. Let Y_1 and Y_2 be the two components of Y. Let us write $f = v_1 v_2$, with $v_i \in Y_i$. The endpoints of f belong to distinct X-arcs, otherwise f would not cross any X-edge. If both endpoints of f belonged to the external X-arcs, then

f would cross all the X-edges and it would have been pruned. Thus at least one endpoint, w.l.o.g. the endpoint v_1, belongs to an internal X-arc, and the other endpoint belongs to a different X-arc. By Lemma 11, Y is an F-cluster and all the active vertices of Y_1 appear in the same X-arc as v_1.

If all the active vertices of Y_2 belonged to the same X-arc as v_2, then the X-edge e would cross all the Y-edges and it would have been pruned. Thus, Y_2 has active vertices in at least two X-arcs. By Lemma 11, this is only possible if the active vertices of Y_2 belong to the two external X-arcs.

To see that the X-edges form at most two bundles, note that if f and f' are two Y-edges that have endpoints in distinct external X-arcs, then every X-edge crosses f or f'. Since by the invariant I2 no Y-edge may cross X-edges of two distinct bundles, we see that X has at most two bundles. □

Definition 8. *Let X_1 be a connected component of an F-cluster X, let Y_1 and Y_2 be the connected components of an F-cluster Y. We say that Y_1 is* above *X_1 (and X_1 is* below *Y_1) if Y_1 has two active vertices u and v which partition the boundary of F into two $\{u, v\}$-arcs in such a way that the vertices of Y_2 on the boundary of F are in a different $\{u, v\}$-arc than the vertices of X_1 on the boundary of F. (Equivalently, X_1 is below Y_1 if the vertices of X_1 in F appear in an internal Y^a-arc whose endpoints belong to Y_1, where Y^a denotes the set of F-active vertices of Y.)*

Lemma 20 implies that if, for a pair of distinct F-clusters X and Y, an X-edge crosses a Y-edge, then one component of X is above a component of Y, while the other component of X is below the other component of Y.

Lemma 21. *Let X and Y be as in Lemma 20. Then each of the two external X-arcs contains exactly one active vertex of Y.*

Proof. Let X_1, X_2, Y_1 and Y_2 be the components of X and Y respectively, and assume without loss of generality that X_1 is above Y_1 while X_2 is below Y_2. Let α and β be the two external X-arcs. We already know from Lemma 20 that each of these two arcs has at least one active vertex from Y_2. Let γ be the internal X-arc containing active vertices of Y_1. Assume for contradiction that α contains two distinct active vertices $u, v \in Y_2$, and fix two Y-edges e and f adjacent to u and v respectively. Let g be an X-edge that intersects both e and f. Such an X-edge g must exist — if there were no such edge, then the Y-edges with endpoints in β and γ would intersect all the X-edges, which is impossible. This also implies that e and f must belong to the same Y-bundle, otherwise the edge g would contradict I2.

Let us choose e and f so that they share a common endpoint $w \in Y_1$. The edges e and f are not equivalent, otherwise one of them would have been pruned by R4. This leaves us with two possibilities:

1. There is a permitted edge h that crosses e but not f (or vice versa).
2. There is a permitted edge h that interferes with e but not with f (or vice versa).

Let us first deal with case 1. Let Z be the cluster containing h. Let us apply Lemma 20 to the two clusters Y and Z. Since h has at least one endpoint in an internal Y-arc whose endpoints are in Y_2, the cluster Z has a component Z_2 which is below Y_2. Then the other component (call it Z_1) must be above Y_1. It follows that at least one Z-edge must cross the X-edge g. Let us apply Lemma 20 to the two clusters X and Z. We see that the active vertices of Z_2 appear in the external X-arc α. Thus, there must also be a vertex of Z_2 in β, which is impossible, since all the active vertices of Z_2 are in the same internal Y-arc.

Let us now deal with case 2. Let Z be the cluster containing h, let W be the smallest common disconnector of Y and Z. Assume that W has components W_1 and W_2, with $Y_i \subseteq W_i$ for $i = 1, 2$. The vertices u and v belong to different active arcs of W_2, otherwise e and f would be equivalent. By Lemma 15, W_2 cannot have more than two active arcs. In particular, W_2 must have an active arc δ that intersects both α and β. Since δ must avoid the vertices from Y_1, we see that δ contains all the active vertices of X_2. This shows that $X_2 \subset W_2$ and $X \subseteq W$. If W is a connector of X, then the edge g is blocked by W (note that all the three vertices u, v and w belong to different $\overline{W}$-arcs) and should have been removed by R2. If W is a disconnector of X, we also get a contradiction, because all the active vertices from X_2 belong to the same active arc δ of W_2, hence the other active arc of W_2 cannot be fully active, contradicting Lemma 15.

We conclude that α contains at most one active vertex of Y, as claimed. $\square$

We now perform a similar analysis for edge-interference.

Lemma 22. *Let X be an F-cluster. Assume that Y is another cluster, and that there is a Y-edge that interferes with an X-edge in a cluster Z. The following holds:*

1. *Y is also an F-cluster.*
2. *Y has at most two bundles.*
3. *If X' is a component of X and Y' is a component of Y such that X' and Y' are subsets of the same component of Z, then X' is either above or below Y'.*
4. *Each component of Y has at most two active vertices.*

Proof. The first claim is a direct consequence of Lemma 15. Let us prove the second claim. Let Z be the smallest common disconnector of X and Y. Let X_1, X_2, Y_1, Y_2, Z_1, and Z_2 be the components of X, Y, and Z, with $X_i \cup Y_i \subseteq Z_i$ for $i = 1, 2$. At least one of the two components of Z must have two active arcs, otherwise no permitted edges from X or Y would interfere. Assume that Z_1 has two active arcs. Then, by invariant I3, all the active vertices of Y (and X) in a given active arc of Z_1 belong to the same bundle, hence Y has at most two bundles.

Let X_i, Y_i and Z_i be as above. We will show that X_i is above or below Y_i for $i = 1, 2$. Let us first deal with X_1 and Y_1. Since Z_1 has two active arcs, each of them containing an active vertex of X_1 as well as from Y_1, and since X_1 and Y_1 cannot cross, we see that X_1 is either above or below Y_1, as claimed.

Let us now deal with X_2 and Y_2. We claim again that X_2 is above or below Y_2. Note that if there is an X-edge that crosses a Y-edge, then the claim follows from Lemma 20. On the other hand, if there is no such crossing, then clearly X_2 is above Y_2 (or below Y_2) whenever X_1 is above Y_1 (or below Y_1, respectively).

Let us argue that each component of Y has at most two active vertices. Assume that a component Y_i of Y has three active vertices u, v and w. Let W be the smallest disconnector of Y that also disconnects another cluster, and let W_1 be the component of W that contains u, v, w. We claim that among the three vertices u, v, w there is a pair of vertices that belong to the same Y-bundle and to the same active arc of W. Indeed, if W_1 has only one active arc, then this claim follows from the fact that Y has at most two bundles. On the other hand, if W_1 has two active arcs, then this claim follows from the fact that all the active vertices in a given arc belong to the same bundle, due to I3. Thus, we may assume that u and v belong to the same bundle and to the same active arc of W_1.

Let e and f be two Y-edges that are incident to u and v respectively and have a common endpoint in W_2. Note that if g is any permitted edge that is not a Y-edge, then g interferes with e if and only if g interferes with f. This is because if Z is a common disconnector of Y and of the cluster containing g, then $W \subseteq Z$ by the minimality of W, and hence u and v belong to the same active arc of Z.

Since e and f are not equivalent, even though they both interfere with the same set of edges, we conclude that there must be an edge h that is not a Y-edge and that crosses exactly one of e and f. Applying Lemma 21, we conclude that the component of Y containing u and v has exactly two active vertices, as claimed. □

Lemma 23. *Let X be an F-cluster. Every component of X has at most two F-active vertices.*

Proof. Assume that there is an F-cluster X with components X_1 and X_2, and assume that X_1 has three active vertices u, v and w. Since every permitted edge participates in a crossing or an interference, we know from Lemma 20 and Lemma 22 that X has at most two bundles. We may thus assume that u and v belong to the same bundle. Let e and f be two X-edges incident to u and v respectively, and assume that e and f have a common endpoint in X_2. Since e and f are not equivalent, there are two possibilities:

1. There is a permitted edge g which is not an X-edge and which crosses exactly one of e and f.
2. There is a permitted edge g which is not an X-edge and which interferes with exactly one of e and f.

In the first case, we get a contradiction with Lemma 20 and Lemma 21, while the second case contradicts Lemma 22. □

From the previous lemma we see that if X is an F-cluster, then the X-edges form one of the configurations depicted in Fig. 3 on page 10. Furthermore, if X

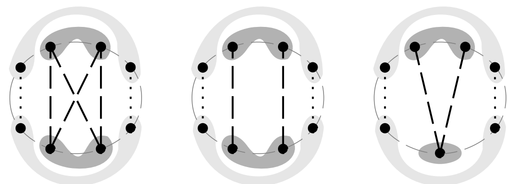

Fig. 17. Possible configurations of permitted edges of two F-clusters X and Y, assuming that no X-edge crosses a Y-edge, but there is an interference of X-edges and Y-edges.

and Y are distinct F-clusters such that an X-edge crosses a Y-edge, then the X-edges and Y-edges together form one of the configurations depicted in Fig. 4 on page 11. Finally, if none of the X-edges crosses a Y-edge but there is an interference between an X-edge and a Y-edge, then these permitted edges form one of the three configurations of Fig. 17.

Let us define a 'symmetry function' σ on the set of F-active vertices of F-clusters as follows: if a component of an F-cluster has only one active vertex v, then $\sigma(v) = v$. If a component of an F-cluster has two active vertices u and v, then $\sigma(u) = v$ and $\sigma(v) = u$. We now extend the mapping σ to the set of all the permitted edges of F-clusters: if $e = uv$ is a permitted edge of an F-cluster, then $\sigma(e)$ denotes the candidate edge with endpoints $\sigma(u)$ and $\sigma(v)$.

Note that e is a permitted edge if and only if $\sigma(e)$ is a permitted edge, and that $\sigma(e) \neq e$. We will say that the two edges e and $\sigma(e)$ are *twins*. An F-cluster has either two or four permitted edges, forming one or two pairs of twins.

Lemma 24. *Let X and Y be two distinct F-clusters, let e be an X-edge, let f be a Y-edge. The edge e crosses f if and only if $\sigma(e)$ crosses $\sigma(f)$. The edge e interferes with f if and only if $\sigma(e)$ interferes with $\sigma(f)$.*

Proof. For edge-crossing, the follows directly from the previous lemmas. Let us deal with the case of edge-interference. Assume that an X-edge e interferes with a Y-edge f. Let Z be the minimal common disconnector of X and Y. Assume that the components of the three clusters satisfy $X_i \cup Y_i \subseteq Z_i$ for $i \in \{1, 2\}$. By Lemma 15, a component of Z may have one or two active arcs in F, and each of its active arcs is fully active. If Z_i has one active arc, then this arc contains all the active vertices of $X_i \cup Y_i$. If Z_i has two active arcs α and β then, in view of Lemma 23, each of these two arcs has exactly one active vertex from X_i and exactly one active vertex from Y_i. The mapping σ then maps an active vertex of α to an active vertex of β and vice versa. It follows that σ preserves edge-interference. □

Lemma 25. *Let X be an F-cluster. Recall that P is the set of permitted edges. If there is a minimal saturator $S \subseteq P$ which contains an X-edge e, then there is also a minimal saturator $S' \subseteq P$ which contains $\sigma(e)$. In particular, if X has only two permitted edges, then they are both harmless.*

Proof. Let $S \subseteq P$ be a minimal saturator containing the X-edge e. Let S' be the set of permitted edges obtained from S by replacing each edge of an F-cluster by its twin, while the edges of S that do not belong to F-clusters are left unchanged in S'. Clearly, S' contains $\sigma(e)$.

We claim that S' is a minimal saturator. It is easy to see that S' saturates all the clusters, and each minimal cluster is saturated by a unique edge. We claim that S' has no pair of crossing or interfering edges. Indeed, Lemmas 20 and 22 show that if a permitted edge e crosses or interferes with a permitted edge of an F-cluster, then e must also belong to an F-cluster. From Lemma 24, we see that edges of S' saturating distinct F-clusters do not cross or interfere. It follows that S' is a saturator.

To prove the last claim, it suffices to point out that if an F-cluster X has two permitted edges, then the two edges are twins, and any saturator $S \subseteq P$ must contain at least one of these two X-edges. $\square$

Definition 9. *We say that a component X_1 of an F-cluster X is* short *if no other component of any F-cluster is below X_1.*

Note that among all the components of F-clusters, at least one is short. If a component of an F-cluster has only one active vertex, then it is automatically short, and if all the components of F-clusters have two active vertices, then the component whose active vertices form the shortest arc of F is clearly short.

Lemma 26. *Assume that every F-cluster has four permitted edges. Let X be an F-cluster with components X_1 and X_2, let $x_1, x_1' \in X_1$ and $x_2, x_2' \in X_2$ be the four active vertices of X. Assume that X_1 is short and that the four active vertices of X appear on the boundary of F in the clockwise cyclic order (x_1, x_1', x_2, x_2'). Then the X-edge x_1x_2 is harmless.*

Proof. Consider the two X-edges $e = x_1x_2$ and $e' = x_1'x_2$. The two edges are not equivalent, otherwise one of them would be pruned by R4. It follows that there is a cluster Y and a Y-edge g such that one of the following two possibilities holds:

1. g crosses exactly one of the two edges e, e'.
2. g interferes with exactly one of the two edges e, e'.

The first case would imply, by Lemma 20, that one component of Y must be below X_1, contradicting the shortness of X_1. Let us deal with the second case. Let Z be a minimal common disconnector of X and Y, let Y_1, Y_2, Z_1 and Z_2 be the components of Y and Z, with $X_i \cup Y_i \subseteq Z_i$.

From Lemma 22 and from the shortness of X_1, we deduce that Y_1 is above X_1. Also, Y_2 is either above or below X_2. If Y_2 were above X_2, then there would be two Y-edges that would cross all the X-edges, and these two Y-edges should have been pruned by R3. We conclude that Y_2 is below X_2. Let $y_1, y_1' \in Y_1$ and $y_2, y_2', \in Y_2$ be the four active vertices of Y, and without loss of generality assume that they appear in the clockwise cyclic order (y_1, y_1', y_2, y_2').

We claim that the two twin X-edges x_1x_2' and $x_1'x_2$ cannot appear in any saturator $S \subseteq P$. Since the Y-edge g interferes with exactly one of the two X-edges $e = x_1x_2$ and $e' = x_1'x_2$, the two vertices x_1 and x_1' belong to distinct active arcs of Z_1, and y_1 is in the same active arc as x_1 while y_1' is in the same active arc as x_1'. Hence, the X-edge x_1x_2' crosses the two Y-edges incident to y_1 and interferes with the two Y-edges incident to y_1'. Since every saturator $S \subseteq P$ must contain a Y-edge, we conclude that such a saturator S cannot contain the X-edge x_1x_2'. The X-edge $x_1'x_2$ is forbidden for the same reason. Thus, every saturator $S \subseteq P$ must contain one of the two twin edges x_1x_2 and $x_1'x_2'$. By Lemma 24, both these edges are harmless. □

With all this preparation, the LOCATE-IN-FACE subroutine is simple: given a face F with at least one F-cluster, the subroutine proceeds in two main steps.

– First, check whether there is an F-cluster that has precisely two permitted edges. If there is such a cluster, then output one of these two edges as a harmless edge and stop.
– If there is no F-cluster with two permitted edges, then all the F-clusters have four permitted edges and four active vertices. Find an F-cluster X that has a short component. Let x, x', x'', x''' be the four active vertices of X, in the cyclic order in which they appear on the boundary of F. Output the X-edge xx'' as a harmless edge and stop.

The correctness of the subroutine follows from Lemma 24 and Lemma 26.

A.7 OUTPUT-ANYTHING

If there is no one-faced cluster, the procedure LOCATE-IN-FACE is not applicable. We will show that in such case, every permitted edge is harmless.

We again need some preparation. Throughout this subsection, we assume that $(G, \mathcal{C})$ is a nice clustered graph, P is the set of permitted edges of G that satisfies the four invariants I1-I4, and that cannot be pruned by any of the four rules R1-R4. Furthermore, we assume that none of the triviality checks is applicable, i.e., each cluster has at least two permitted edges, and every permitted edge crosses or interferes with a permitted edge of another cluster. We further assume that the procedure LOCATE-IN-FACE is not applicable, i.e., none of the clusters is one-faced.

We first show that in such situation, each cluster is two-faced.

Lemma 27. *Each cluster is two-faced.*

Proof. Assume for contradiction that there is a cluster X, whose permitted edges appear in at least three distinct faces. Choose a face F that contains an X-edge $e \in P$. By assumption, e must cross or interfere with a permitted edge f that saturates a minimal cluster $Y \neq X$. If the edges e and f interfere, we obtain an immediate contradiction with Lemma 15. Thus, we may assume that the two edges cross.

Since no cluster is one-faced, there must exist a face F' different from F that contains a Y-edge f'. Since the cluster X is many-faced, there must exist a face F'' different from F and F' which contains an X-edge e'.

Consider the graph $G' = G \cup \{e', f'\}$. The graph G' is a plane graph and the two sets X and Y induce connected subgraphs of G'. However, the two sets X and Y cross on the boundary of the face F of G' (due to the two permitted edges e and f), which contradicts Lemma 7.

We conclude that every cluster is two-faced. □

Lemma 28. *Let X be a minimal cluster, and assume that every X-edge belongs to one of the two faces F and F'. Let Y be another minimal cluster, and assume that there is an X-edge that crosses a Y-edge. The following holds:*

1. *Every Y-edge belongs to one of the two faces F and F'.*
2. *If an X-edge and a Y-edge belong to the same face then they cross.*

Proof. Assume that an X-edge e crosses a Y-edge f. Let F be the face that contains the two edges. By assumption, there is an X-edge e' in the face F'. If Y has a permitted edge f' in a face F'' different from F and F', we obtain the same contradiction as in the proof of Lemma 27. This shows that every Y-edge belongs to one of the two faces F and F'.

In fact, this argument shows that all the X-edges in F' intersect all the Y-edges in F', since if there were an X-edge e' and a Y-edge f' in F' that do not cross, then the graph $G \cup \{e', f'\}$ would again contradict Lemma 7. Since F' contains at least one X-edge and at least one Y-edge, we may apply the same argument symmetrically to show that all the X-edges in F cross all the Y-edges in F, as claimed. □

Lemma 29. *Let X be a minimal cluster, and assume that every X-edge belongs to one of the two faces F and F'. Let Y be another minimal cluster, and assume that there is an X-edge that interferes with a Y-edge. The following holds:*

1. *Every Y-edge belongs to one of the two faces F and F'.*
2. *An X-edge and a Y-edge interfere with each other if and only if they belong to different faces.*

Proof. The first claim follows from Lemma 15 and Lemma 13. For the second claim, first observe that X and Y have a common disconnector Z (otherwise there could be no interference between an X-edge and a Y-edge), and hence any X-edge interferes with a Y-edge from a different face. On the other hand, by Lemma 15, each component of Z has at most one active arc in each of the two faces F and F', and hence no two edges in the same face may interfere. □

Corollary 1.

– *For a minimal cluster X, all the X-edges that belong to the same face are equivalent.*

– *Let X and Y be distinct minimal clusters. Let e and e' be two X-edges that belong to different faces, let f and f' be two Y-edges that belong to different faces. Then e crosses f if and only if e' crosses f', and e interferes with f if and only if e' interferes with f'.*

Proof. This is a direct consequence of Lemma 28 and Lemma 29. □

This brings us to the main lemma of this subsection.

Lemma 30. *Every permitted edge is harmless.*

Proof. Let $e \in P$ be an X-edge. Let us prove that e is harmless. Let $S \subseteq P$ be a minimal saturator. Let e' be the edge of S that saturates X. If $e = e'$, we are done, so assume that $e \neq e'$.

If e' belongs to the same face as e, then by the first part of Corollary 1 we see that $S \cup \{e\} \setminus \{e'\}$ is a saturator and we are done. On the other hand, if e and e' belong to distinct faces, then we replace the edge e' by e, and simultaneously we replace every other edge $f' \in S$ by an edge f which saturates the same minimal cluster as f' and belongs to a different face. Let S' be the set of edges obtained by this replacement. By the second part of Corollary 1, S' is a saturator. □

This completes the description of the FIND-EDGE algorithm. As in the flat case, it is clear that the algorithm runs in polynomial time.