Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 22, no. 2, pp. 239–271 (2018)
DOI: 10.7155/jgaa.00466

Recognizing IC-Planar and NIC-Planar Graphs

Franz J. Brandenburg

94030 Passau, Germany

Abstract

We prove that triangulated IC-planar graphs and triangulated K_5-free or $X4W$-free NIC-planar graphs can be recognized in cubic time. A graph is 1-planar if it can be drawn in the plane with at most one crossing per edge. A drawing is IC-planar if, in addition, each vertex is incident to at most one crossing edge and NIC-planar if two pairs of crossing edges share at most one vertex. In a triangulated drawing each face is a triangle. A graph is K_5-free ($X4W$-free) if it does not contain simple K_5 with a separating 3-cycle (extended 4-wheel graphs). In consequence, planar-maximal and maximal IC-planar graphs can be recognized in $O(n^5)$ time and maximum and optimal ones in $O(n^3)$ time.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
October 2016	February 2017	August 2017	April 2018	April 2018
		Published:		
		May 2018		
	Article type:		Communicated by:	
	Regular paper		A. Symvonis	

E-mail address: brandenb@informatik.uni-passau.de (Franz J. Brandenburg)

1 Introduction

Graphs are commonly drawn in the plane so that the vertices are mapped to distinct points and the edges to Jordan curves connecting the endpoints. A drawing is used to visualize structural relationships that are modeled by vertices and edges and thereby make them easier comprehensible to a human user. Specifications of nice drawings of graphs and algorithms for their construction are the topic of Graph Drawing [18, 26, 33].

There are several classes of graphs that are defined by specific restrictions of edge crossings in graph drawings. Edge crossings are negatively correlated to nice, and therefore, they should be avoided or controlled in some way. The planar graphs are the best known and most prominent example. Planarity excludes crossings and is one of the most basic and influential concepts in Graph Theory. Many properties of planar graphs have been explored, including duality, minors, and drawings [20], as well as linear-time algorithms for the recognition and the construction of straight-line grid drawings [33]. However, graphs from applications in engineering, social science, and life science are generally not planar. This observation has motivated approaches towards *beyond-planar* graphs, which allow crossings of edges with restrictions. A prominent example is 1-planar graphs, which were introduced by Ringel [31] in an approach to color a planar graph and its dual simultaneously. A graph is *1-planar* if it can be drawn in the plane so that each edge is crossed at most once. 1-planar graphs have found recent interest [27]. A 1-planar graph of size n has at most $4n-8$ edges [31] and K_6 is the maximum complete 1-planar graph. Not all 1-planar graphs admit straight-line drawings [19, 34], whereas 3-connected 1-planar graphs can be drawn straight-line on a grid of quadratic size with the exception of a single edge in the outer face [1]. Moreover, not all 1-planar graphs admit right angle crossing drawings [21], and conversely, there are right angle crossing (RAC) graphs that are not 1-planar. In other words, the classes of 1-planar and RAC graphs are incomparable. The recognition problem of 1-planar graphs is NP-complete [24, 28]. It remains NP-complete, even for graphs of bounded bandwidth, pathwidth, or treewidth [6], if an edge is added to a planar graph [13], and if the graphs are 3-connected and are given with a rotation system which describes the cyclic ordering of the neighbors at each vertex [4]. On the other hand, 1-planar graphs can be recognized in cubic time if they are *triangulated* and admit a drawing with triangular faces [16] and even in linear time if they are *optimal*, i.e., if they have $4n-8$ edges [10].

1-planar graphs can also be defined in terms of maps [14, 15, 16, 35]. Maps generalize the concept of planar duality. A *map* M is a partition of the sphere into finitely many regions. Each region is homeomorphic to a closed disk and the interior of two regions is disjoint. Some regions are labeled as *countries* and the remaining regions are lakes or *holes*. In the plane, we use the region of one country as the outer face, which is unbounded and encloses all other regions. An *adjacency* is defined by a touching of countries. There is a *strong* adjacency between two countries if their boundaries intersect in a segment and a *weak* adjacency if the boundaries intersect only in a point. There is a k-point if k

countries meet at a point. A map M defines a graph G so that the countries of M are in one-to-one correspondence with the vertices of G and there is an edge $\{u, v\}$ if and only if the countries of u and v are adjacent. Then G is called a *map graph* and M is the map of G. Note that holes are discarded for the definition of map graphs. Obviously, a k-point induces K_k as a subgraph. If no more than k countries meet at a point, then M is a k-map and G is a *k-map graph*. If there are no holes then M is hole-free. A graph is a *hole-free 4-map graph* if it is the map graph of a hole-free 4-map [14, 15, 16].

Chen et al. [15] observed that 4-map graphs are 1-planar and stated [16] that the triangulated 1-planar graphs are exactly the 3-connected hole-free 4-map graphs. Their main result in [16] is a cubic-time recognition algorithm for 3-connected hole-free 4-map graphs. They claimed a cubic time recognition algorithm for 4-map graphs [14] and refer to a draft for a proof. Recently, the author characterized hole-free 4-maps graphs as kite-augmented 1-planar graphs and established a cubic time recognition algorithm [9]. 1-planar graphs are kite-augmented if they admit a 1-planar drawing with a K_4 induced by the endvertices of each pair of crossing edges. In this work, we extend the algorithm by Chen et al. [16] to triangulated 1-planar graphs with (near) independent crossings.

A graph is IC-*planar* (independent crossing planar) [2, 11, 29, 37] if it has a 1-planar drawing in which each vertex is incident to at most one crossing edge and is NIC-*planar* (near independent crossing planar) [5, 36] if two pairs of crossing edges share at most one vertex. If each pair of crossing edges is augmented to the complete graph K_4, which can be drawn as a *kite* [31] as in Fig. 1(b), then a 1-planar drawing is IC-planar if each vertex is part of at most one kite and it is NIC-planar if each edge is part of at most one kite.

It is known that IC-planar graphs have at most $13/4\,n - 6$ edges [29] and are 5-colorable [37]. The recognition problem is NP-hard, even for 3-connected graphs with a given rotation system [11]. IC-planar graphs admit straight-line drawings on a grid of quadratic size and right angle crossing drawings, which, however, may need exponential area [11]. Hence, every IC-planar graph is a RAC graph. NIC-planar graphs have at most $18/5\,(n-2)$ edges [36] and this bound is tight for all $n = 5k + 2$ and $k \geq 2$ [5]. The recognition problem is NP-complete, whereas optimal NIC-planar graphs with exactly $18/5\,(n-2)$ edges can be recognized in linear time [5]. NIC-planar graphs admit straight-line drawings, but not necessarily with right angle crossings. In fact, there are NIC-planar graphs that are not RAC graphs, and vice-versa [5]. Hence, the classes of NIC-planar and RAC graphs are incomparable. Outer 1-planar graphs are another important subclass of 1-planar graphs that admit 1-planar drawings with all vertices in the outer face [22]. Outer 1-planar graphs are planar [3] and can be recognized in linear time [3, 25].

In this work we extend the cubic-time algorithm of Chen et al. [16] for the recognition of triangulated 1-planar graphs first by an edge coloring and then to triangulated IC-planar and NIC-planar graphs. We call the algorithms $\mathcal{A}$, $\mathcal{B}, \mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$, respectively. Our algorithms are presented as programs. They compute an edge coloring, that classifies edges as non-crossed, crossed

Figure 1: Drawings of K_4 (a) planar as a tetrahedron and (b) with a crossing as a kite.

or possibly crossed and a boolean formula, which is used to test for IC- and NIC-planarity. An edge is possibly crossed if it is crossed in one embedding and non-crossed in another. Such edges occur in a K_5 (Fig. 3), which admits six embeddings if the outer triangle is fixed, in 4^+-wheel graphs (Fig. 5), and at separating edges (Fig. 4). IC-planarity can be tested efficiently, whereas in the NIC-planar case we need further restrictions, namely no K_5 or no X4W-graphs, which are defined in Section 6.

The paper is organized as follows. Section 2 describes basic definitions. In Section 3 we recall the algorithm by Chen et al. [16] and present our extension in Section 4. In Section 5 we specialize the algorithm to IC- and NIC-planar graphs and analyse their running time in Section 6. We conclude in Section 7 with some open problems.

2 Preliminaries

We consider undirected graphs $G = (V, E)$ and assume that the graphs are simple and 2-connected, unless otherwise stated. The subgraph induced by a subset U of vertices is denoted by $G[U]$. For convenience, we omit braces and write $G[u_1, \ldots, u_r]$ if $U = \{u_1, \ldots, u_r\}$. The subgraph of G induced by the vertices of subgraphs H and K of G is denoted by $H + K$. Similarly, let $G - H$ denote the subgraph induced by the vertices of G without the vertices of H, except if H is a set of edges, which are removed from G, whereas their endvertices remain.

A *drawing* of G maps the vertices to distinct points in the plane and each edge $\{u, v\}$ to a Jordan curve that connects the points of u and v and does not pass through other endpoints. Two edges *cross* if their Jordan curves intersect in a point other than an endpoint. A *planar* drawing excludes edge crossings and a *1-planar* drawing admits at most one crossing per edge. A single crossing subdivides an edge into two *half-edges*. An *embedding* $\mathcal{E}(G)$ is an equivalence class of drawings and specifies edge crossings and faces. The boundary of each face in a 1-planar embedding consists of edges between two vertices and of half-edges between a vertex and a crossing point. It can be described by the vertices and crossing points of the boundary. The *planarization* of an embedding $\mathcal{E}(G)$ is an embedded planar graph which is obtained by taking each crossing point as a

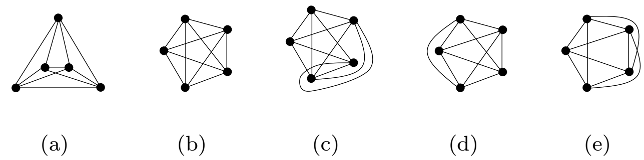


Figure 2: All non-isomorphic embeddings of K_5

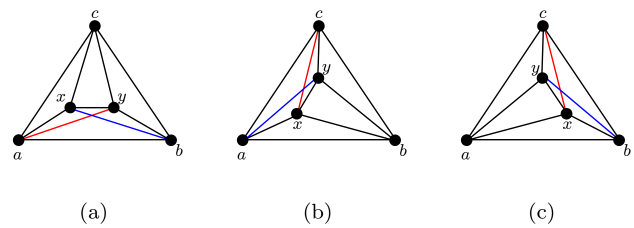


Figure 3: Six embeddings of K_5 with a fixed outer face, where x and y can be exchanged. Each kite includes the edge $\{x, y\}$ and one of the outer edges.

new vertex and half-edges as new edges. It is used for an algorithmic treatment of embeddings. Now all faces are triangles if $\mathcal{E}(G)$ is triangulated.

The complete graph on four vertices K_4 plays a crucial role in 1-planar, IC-planar and NIC-planar graphs, respectively. It admits two embeddings up to isomorphism [30], as a *tetrahedron* or as a *kite* with a pair of crossing edges, see Fig. 1. The embedding as a tetrahedron T is not necessarily planar. An edge e of T can be crossed and due to the triangulation it is *covered* by a kite so that e is a crossing edge of the kite, see Figs. 6(a) and 6(b). The cubic-time recognition algorithm for hole-free 4-map graphs of Chen et al. [16] searches all K_4 subgraphs κ of the given graph and checks whether κ must be embedded as a kite or as a tetrahedron. This can be determined to a large extend, but it is not unique, as K_5 illustrates. The complete graph K_5 has five embeddings up to isomorphism [30], as displayed in Fig. 2, but only one of them is 1-planar. If the outer face is fixed, then there are six 1-planar embeddings with one of the outer edges in a kite and an interchange of the two interior vertices x and y, see Fig. 3. Such embeddings of K_5 are called *simple* and they are non-simple if further edges are crossed. We call a graph K_5-*free* if it admits an embedding without simple K_5. Simple K_5 play a special role and are exempted e.g., from the coloring scheme.

3 Recognizing Triangulated 1-planar Graphs

For the recognition of triangulated IC-planar and NIC-planar graphs, we extend algorithm $\mathcal{A}$ by Chen et al. [16]. Recall that the 3-connected hole-free 4-map graphs are exactly the triangulated 1-planar graphs. Our algorithm $\mathcal{B}$ extends $\mathcal{A}$ by an edge coloring and a boolean formula. Algorithm $\mathcal{A}$ marks an edge if it is non-crossed at the actual stage of the algorithm. It could have been crossed at an earlier stage, in which case it is crossed in the computed 1-planar embedding. This divergence is due to the fact that $\mathcal{A}$ removes one crossing edge if it detects a pair of crossing edges and marks the remaining one as non-crossed. Our edge coloring records each decision and tells whether an edge is crossed or non-crossed in every triangulated 1-planar embedding, or whether this is uncertain and depends on the particular embedding. The uncertainty is expressed by a boolean formula, such that there is a one-to-one correspondence between feasible embeddings and truth assignments if small graphs are restricted to their outer face. An embedding is feasible if it is triangulated and IC- or NIC-planar, respectively. The IC-extension (NIC-extension) describes IC-planarity (NIC-planarity), i.e., that each vertex (edge) is incident to at most one crossing edge. We prove the following:

Theorem 1 *There is a cubic-time algorithm that checks whether a graph G is a triangulated 1-planar graph. It returns an edge coloring of G, which tells whether an edge is crossed (non-crossed) in every 1-planar embedding or whether this is unclear and depends on the embedding, and a boolean formula η such that the IC-extension (NIC-extension) η^+ of η is satisfiable if and only if G is IC-planar (NIC-planar). Otherwise, the algorithm returns* false *and stops with a failure.*

Algorithm $\mathcal{B}$ is the program of algorithm $\mathcal{A}$ of [16] with some minor modifications. Algorithms $\mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$ specialize $\mathcal{B}$ to IC-planar and NIC-planar graphs, respectively. These algorithms operate on 3-connected edge-colored graphs, whereas algorithm $\mathcal{A}$ does not distinguish black, grey, brown, blue, and cyan edges and just marks them. The edge coloring is defined in Section 4. In addition, algorithms $\mathcal{B}$, $\mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$ construct a boolean formula. Next, we describe the steps of the algorithms, the edge coloring, and the boolean formulas.

Algorithm $\mathcal{A}$ "makes progress" (i) by a separation into 4- and 5-connected components and (ii) by a crossing removal. It systematically checks all K_4 subgraphs κ of the given graph G and tries to determine whether κ must be embedded as a kite or as a tetrahedron, and so do algorithms $\mathcal{B}$, $\mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$. In most cases there is an unambiguous decision. Ambiguities are expressed by the edge coloring and the boolean formula. For example, the complete graph K_5 with a fixed outer face has three embeddings up to isomorphism with kites κ_1, κ_2, and κ_3, as illustrated in Fig. 3, and the clause $(e^1_{\kappa_1} \vee e^2_{\kappa_2} \vee e^3_{\kappa_3})$ expresses the three options, where e^i corresponds to a pair of vertices (edge) in the outer face.

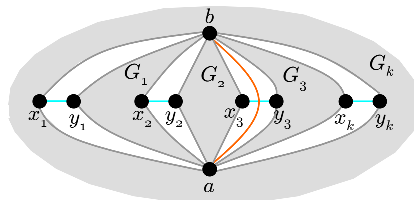


Figure 4: A separating edge $\{a, b\}$; the shaded areas represent 4-connected subgraphs. The orange edge $\{a, b\}$ must cross one of the cyan ones, which are separating crossable edges $\{x_i, y_i\}$ for $i = 1, \ldots, k$ and $k \geq 2$.

Definition 1 *Let G be a 3-connected graph with some colored edges. For an uncolored edge $\{a, b\}$ let $\mathcal{C}[a, b]$ be the set of uncolored edges $\{x, y\}$ so that the induced subgraph $G[a, b, x, y]$ is K_4 and is not included in a K_5 subgraph $G[a, b, x, y, z]$. Such edges are called* crossable edges *in* [16].

If there are edges $\{a, b\}$ and $\{x, y\}$ with $\{x, y\} \in \mathcal{C}[a, b]$, then both edges are candidates for a crossing. They are uncolored and their endvertices induce K_4, which is maximal and is not part of a K_5. Edges of K_5 subgraphs κ are discarded, except if they are part of another maximal K_4 with vertices not in κ.

The algorithms use the following separators:

Definition 2 *Let G be a 3-connected graph.*

1. *A* separating 3-cycle $C = (a, b, c)$ *is a 3-cycle such that $G - C$ is disconnected and partitions into G_{in} and G_{out}.*
2. *A* separating edge *is an uncolored edge $\{a, b\}$ such that $G - \{a, b\} - \mathcal{C}[a, b]$ is disconnected, see Fig. 4. An edge $\{x, y\} \in \mathcal{C}[a, b]$ is called a* separating crossable edge *if x and y are in different components of $G - \{a, b\} - \mathcal{C}[a, b]$.*
3. *A* separating triple *is a 3-cycle $C = (a, b, c)$ such that $G - C - \mathcal{C}[a, b]$ is disconnected, see Fig. 6(a).*
4. *A* separating 4-cycle $C = (a, b, c, d)$ *is a 4-cycle such that $G - C$ is disconnected and partitions into G_{in} and G_{out}.*
5. *A* separating triangle *is a 3-cycle $C = (a, b, c)$ such that $G - C - \mathcal{C}[a, b] - \mathcal{C}[b, c]$ is disconnected, see Fig. 6(b).*
6. *A* separating quadruple *$C = (a, b, c, d)$ is a 4-cycle such that $G - C - \mathcal{C}[a, b]$ is disconnected.*

Algorithms $\mathcal{B}$, $\mathcal{B}_{IC}$, and $\mathcal{B}_{NIC}$ use these separators in the prescribed order. Hence, there is no separating 3-cycle if a separating edge (or any other separator) is searched. Then 3-connected triangulated 1-planar graphs are 4-connected

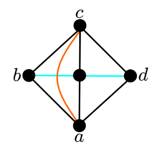


Figure 5: An extended 4^+-wheel graph is the planar $K_5 - e$ and is embedded with an outer 4-cycle and an edge $\{a, c\}$ that crosses one of two possible edges.

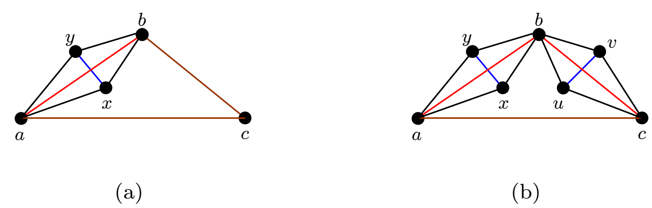


Figure 6: (a) A separating triple with a kite-covered edge $\{a, b\}$ and (b) a separating triangle with two kite-covered edges.

as proved in [16]. Similarly, the graphs are 5-connected if there are no separating 4-cycles and separating triangles or quadrangles are searched. The set of separating crossable edges $\mathcal{C}[a, b]$ consists of a single edge in a separating triple, triangle or quadruple.

Chen et al. [16] propose to use separating 4-cycles before separating triples and separating quadruples before separating triangles. However, this is useless, since a separating triangle implies a separating 4-cycle and a separating triangle implies a separating quadruple. In their correctness proof, Chen et al. do not rely on the order and use only the 4-connectivity of graphs.

If none of the separators applies, and the graph has size at least nine and contains a K_4, then algorithm $\mathcal{A}$ shall first search for K_5 subgraphs, called MC_5, and if there are no K_5, it considers (the remaining) K_4 subgraphs and determines whether they must be embedded as a tetrahedron or as a kite, called MC_4. However, at this stage, the graphs are 5-connected and the absence of the above separators implies that there are no K_5 subgraphs or the graphs are small [9]. In consequence, the MC_5 step described in [16] can be skipped and we must consider embeddings of K_5 only as part of small subgraphs of size at most eight that are obtained by separating 3- or 4-cycles.

Finally, in the MC_4 step, a detected K_4 must be embedded as a tetrahedron if all its edges are crossed and the endvertices of the crossing edges are distinct such that there is a completely kite-covered tetrahedron, as shown in Fig. 7(a), or there is an SC-graph, as shown in Fig. 7(b), in which the edges of the outer cycle of the tetrahedron are crossed by edges incident to its forth (inner) vertex. Otherwise, the detected K_4 subgraph must be embedded as a kite. Recall that a

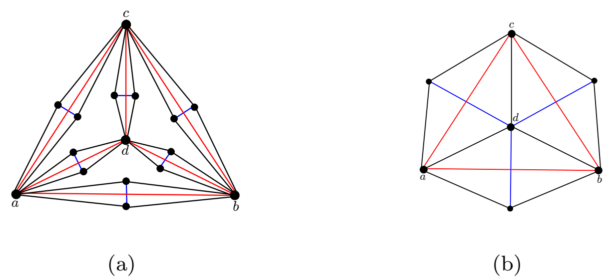


Figure 7: A (a) completely kite-covered tetrahedron and (b) an SC-graph

planar tetrahedron is detected in the first step, since there is a separating 3-cycle. If some endvertices of crossing edges in completely kite-covered tetrahedrons coincide, then there are separating triples, triangles, or quadrangles.

In the "make progress" steps algorithm $\mathcal{A}$ proceeds as follows: If a separating 3-cycle or 4-cycle C partitions $G - C$ into G_{in} and G_{out}, then $\mathcal{A}$ recursively proceeds on $G_{in} + C$ and $G_{out} + C$. It merges the computed embeddings and thereby identifies the edges of C. If C is a 4-cycle, then a chord f must be added to the subgraphs for a triangulation, and f must be chosen properly, such that it is new for the remaining subgraph. The chord is removed if the subgraphs are merged later on. At the other separators $\mathcal{A}$ removes edges that must be crossed and destroys all detected K_4 that must be embedded as a kite. For example, it removes one edge from the pair of crossing edges in a separating triangle, see 6(b). Which one does not really matter. However, the removed edge has an impact on the computation process and the edge coloring. For example, if there is a separating triple so that edges $\{a, b\}$ and $\{x, y\}$ cross, then there is a separating 3-cycle (a, b, c) if $\{x, y\}$ is removed and there is a separating 4-cycle (a, b, c, y) if $\{a, b\}$ is removed and $\{a, b\}$ is reinserted as a chord, and thereafter, there is a separating 3-cycle (a, b, x) that splits off vertex y. Similarly, $\mathcal{A}$ destroys all detected kites in the MC_4 step and, for example, removes six edges of a completely kite-covered tetrahedron and three edges from an SC-graph, see Fig. 7. Plain kites are common in IC- and NIC-planar graphs and they are surrounded by planar subgraphs, as the study of maximal NIC-planar graphs shows [5].

We use the following properties of algorithm $\mathcal{A}$, which were proved by Chen et al. [16] using maps. A simpler proof using embeddings of 1-planar graphs is given in [9].

Lemma 1 *Let G be a 3-connected triangulated 1-planar graph with $|G| > 8$.*

1. *The edges of separating 3-cycles and 4-cycles are non-crossed at the time of their detection and G is partitioned into two components.*

2. *If $\{a, b\}$ is a separating edge, then $\{a, b\}$ and exactly one of its separating crossable edges are crossed in every 1-planar embedding. The edges*

$\{a, x\}, \{a, y\}, \{b, x\}$ and $\{b, y\}$ are non-crossed if $\{x, y\}$ is a separating crossable edge.

3. *Edge $\{a, b\}$ of a separating triple (quadruple) is crossed in every 1-planar embedding, and similarly edges $\{a, b\}$ and $\{b, c\}$ of a separating triangle.*

4. *If $\{x, y\}$ is a crossable edge of $\{a, b\}$ in a separating triple, then $\{x, y\}$ is crossed, whereas $\{a, x\}, \{a, y\}, \{b, x\}, \{b, y\}$ are non-crossed in every 1-planar embedding, and similarly for separating triples and separating quadruples.*

The search for maximal complete subgraphs of size five and four completes algorithm $\mathcal{A}$. However, as stated before, K_5 subgraphs have a unique 1-planar embedding up to isomorphism and the used separators exclude K_5 subgraphs, even for 1-planar graphs [9]. For NIC-planar (IC-planar) graphs the arguments are even simpler.

Lemma 2 *If G is a triangulated NIC-planar (IC-planar) graph, then the MC_5 step of algorithm $\mathcal{A}$ is vacuous.*

Proof: A K_5 has a unique 1-planar embedding up to isomorphism [30] as shown in Fig. 2(a). At this stage of algorithm $\mathcal{A}$ the graphs are 5-connected. If a, b, c are the vertices in the outer face and $\{a, y\}$ crosses $\{b, x\}$ as in Fig. 3(a), then $\{a, c\}$ and $\{b, c\}$ are crossed by 5-connectivity. Suppose $\{u, v\}$ crosses $\{a, c\}$ for some vertices u, v. Since there are no separating triples or separating triangles, edge $\{a, c\}$ must be part of a K_5 π so that $\mathcal{C}[a, c]$ is undefined. Since π includes the vertices a, c, u, v and some vertex z we have $u = x$ and $z = b$ or $z = y$, since a 1-planar embedding of π is impossible, otherwise. Then $\{a, x\}$ is incident to two K_4 that are embedded as a kite, i.e., to two pairs of crossing edges, a contradiction to NIC-planarity (IC-planarity). □

As proved in Section 9.1 of [16], the remaining K_4 subgraphs are embedded as a tetrahedron if they are completely kite-covered tetrahedrons or SC-graphs. Otherwise, they are embedded as a kite. The algorithms first search for the tetrahedrons, and if there are none, they pick any K_4 κ and embed it as a kite. The cyclic order (a, b, c, d) of the vertices of κ can be computed from the local environment [16]. Note that the cyclic order is determined if a vertex of κ is incident to two colored edges in κ. Alternatively, κ can be contracted to a single vertex and the cyclic order is obtained from the final planar embedding [5, 9].

Lemma 3 *[16] If MC_4 applies to algorithm $\mathcal{A}$, then a detected K_4 subgraph is a completely kite-covered tetrahedron or an SC-graph and if neither of these cases applies, then it must be embedded as a kite.*

Finally, algorithm $\mathcal{A}$ checks whether the obtained graph is a triangulated planar graph or is a small triangulated 1-planar graph. It makes at most $O(n)$ "make progress" steps. The search and test of each separator takes $O(n^2)$ time. The use of MC_4 and the a final check of planarity take linear time, so that the total running time is $O(n^3)$ [16].

4 Recognizing IC- and NIC-Planar Graphs

In each step algorithm $\mathcal{A}$ finds edges that are crossed or are non-crossed in every triangulated 1-planar embedding and it removes one edge from a pair of crossing edges to make progress towards planarity. However, the decisions of $\mathcal{A}$ do not uniquely determine a 1-planar embedding. For example, a separating edge has the choice among several crossable edges. The case resembles a graph decomposition at a separation pair with an arbitrary order of the components. Also, K_5 has six embeddings, which are isomorphic in pairs if there is a planar outer 3-cycle. IC- and NIC-planarity need more information on all 1-planar embeddings of the given graph, which is provided by an edge coloring and a boolean formula.

Algorithm $\mathcal{B}$ is the program of algorithm $\mathcal{A}$ [16] with some minor modifications. Algorithms $\mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$ specialize $\mathcal{B}$ to IC-planar and NIC-planar graphs, respectively. They stop immediately if a violation of IC- or NIC-planarity is detected.

Definition 3 *An edge of a 1-planar graph G that is not an inner edge of a simple K_5 subgraph of G is colored* black *if it is a non-crossed edge of a kite in every triangulated 1-planar embedding $\mathcal{E}(G)$. Other non-crossed edges are colored* grey *or* brown*. Two edges e and f are colored* red *and* blue*, respectively, if e and f cross in every triangulated 1-planar embedding. An edge is colored* orange *if it is crossed and the candidates for a crossing are colored* cyan.

The edges of a simple K_5 are exempted from the coloring scheme, since the pair of crossing edges is unclear, see Fig. 3. We could color them with a different color. For convenience, we choose one edge between an outer and an inner vertex and color it red or orange and apply the coloring scheme otherwise.

Algorithm $\mathcal{A}$ and similarly algorithms $\mathcal{B}, \mathcal{B}_{NIC}$ and $\mathcal{B}_{IC}$ determine that an edge is non-crossed if it is part of a separating 3-cycle or 4-cycle. It is first colored brown and may later be colored black. A separating edge $\{a, b\}$ is crossed and it has the choice to cross any of its separating crossable edges. Note that not all edges of $\mathcal{C}[a, b]$ must be separating crossable edges. Edge $\{a, b\}$ is colored orange and its separating crossable edges are colored cyan. The non-crossed edges from the possible kites are first colored grey. There is a pair of crossing edges in a separating triple, one of which is colored red and the other is colored blue. For the check of IC- and NIC-planarity it is preferable to color edge $\{a, b\}$ red. The edges from the resulting kite are colored black, even if they were colored brown or grey before, and the other edges of the triple are colored brown, except if they were colored in earlier steps. The case of separating triangles and separating quadruples is similar. In the MC_4 step, there are six detected pairs of crossing edges in a completely kite-covered tetrahedron, three such pairs in an SC-graph and one pair in a kite. In each case one crossed edge is colored red and the other is colored blue and the remaining edges of each kite are colored black. The coloring scheme is also applied to small graphs with an outer 3- or 4-cycle. Edges that are not part of a K_4 or are part of a final planar subgraph are colored brown.

The black, grey, brown, red, blue, and orange edges (except in simple K_5) are decided, whereas a cyan edge may be non-crossed in one 1-planar embedding and crossed in another. Black edges are part of a kite, grey edges of possible kites and brown edges are not part of kites at all. A partial coloring χ on a subset of edges of G is extended step by step such that some uncolored edges are colored, where grey and brown edges may be colored black.

Definition 4 *A coloring γ of a set of edges $F \subseteq E$ extends a partial edge coloring χ of G if colored edges keep their color except if a grey or brown edge shall be colored black and an uncolored edge $e \in F$ takes the color of γ.*

The edge coloring extends marked edges by Chen et al. [16] such that black, grey, brown, blue, and cyan edges are marked. The edge coloring may be use to detect errors, for example if a black, grey or brown edge shall be colored blue or red.

In each step, the algorithms $\mathcal{B}, \mathcal{B}_{NIC}$ and $\mathcal{B}_{IC}$ add a term to a formula η or they combine two formulas. The formulas are a conjunction of terms and are almost in CNF. The boolean formulas for IC-planar and NIC-planar graphs have the same structure, however, the boolean variables and the evaluation are different. For every K_4 subgraph κ of the given graph G that may be embedded as a kite, the clause $\alpha(\kappa) = (a_\kappa \wedge b_\kappa \wedge c_\kappa \wedge d_\kappa)$ expresses this fact and the boolean variable x_κ is assigned the value `true` if κ is embedded as a kite. The embedding is not clear if there is a separating edge. Then every possible candidate $G[a, b, x, y]$ is taken into account, where $\{a, b\}$ is a separating edge and $\{x, y\}$ a separating crossable edge. The K_4 subgraph with a pair of crossing edges is determined if there is a separating triple, triangle, or quadruple and in the MC_4 step, and its edges are colored black, red, and blue, respectively. The boolean variable x_κ is associated with vertex x of κ in the IC-planar case, and with the black or grey edge of κ in the NIC-planar case. Feasibility is granted by IC- and NIC-*extensions* of the form $(\neg x_\kappa \vee \neg x_{\kappa'})$ for every vertex (edge) x and kites κ and κ' that may include x.

The subroutines *tri* and *quad* operate as in the case of a separating triple. The non-crossed edges of each kite are colored black, the crossed edges $\{a, b\}$ and $\{a, c\}$ are colored red and their crossed edges blue. The further non-crossed edges are colored brown. They add a clause $\alpha(\kappa)$ to η for each detected kite κ and return $G - \{a, b\}$ $(G - \{\{a, b\}, \{a, c\}\})$ and η.

The subroutine *merge* reverts the partition into an inner and an outer subgraph and inserts the embedding of $G_{in} + C$ in the face left by C in $\mathcal{E}(G_{out} + C)$, identifying the edges of C. It ignores the chord that was added for the triangulation if this was a partition by a separating 4-cycle. It takes the edge coloring of the subgraphs and colors edges of C black if they are black in one of $G_{in} + C$ and $G_{out} + C$. Finally, it combines the boolean formulas by a conjunction.

Algorithm *final-check* determines whether the given graph G is a triangulated planar graphs and returns a planar embedding of G if $|G| > 8$. For small graphs G it determines whether G has a 1-planar embedding extending the

Algorithm 1: Algorithm $\mathcal{B}$

Input: A 3-connected graph G with a partial edge coloring and a boolean formula η.
Initially, all edges are uncolored and $\eta = \mathsf{true}$.

Output: An embedding of a planar subgraph of G, an edge coloring, and (an extension of) η.

Color all edges brown that are not part of a K_4;
while *there is a* K_4 *subgraph and* $|G| \geq 9$ **do**
    if *there is a separating 3-cycle* $C = (a, b, c)$ *with* $G - C = \{G_{in}, G_{out}\}$ **then**
        extend the coloring by brown edges for $\{a, b\}, \{b, c\}, \{c, a\}$;
        return merge($\mathcal{B}(G_{in} + C, \eta_{in})$, $\mathcal{B}(G_{out} + C, \eta_{out})$);
    else if *there is a separating edge* $\{a, b\}$ **then**
        extend the coloring by an orange edge $\{a, b\}$;
        set $\xi = \mathsf{false}$;
        foreach *separating crossable edge* $\{x, y\} \in \mathcal{C}[a, b]$ **do**
            extend the coloring by grey edges $\{a, x\}, \{x, b\}, \{b, y\}, \{a, y\}$ and color $\{x, y\}$ cyan;
            set $\xi = \xi \vee \alpha(G[a, b, x, y])$;
        return $\mathcal{B}(G - \{a, b\}, \eta \wedge \xi)$;
    else if *there is a separating triple* $C = (a, b, c)$ *with* $\mathcal{C}[a, b] = \{\{x, y\}\}$ **then**
        color $\{a, b\}$ red, $\{x, y\}$ blue and $\{a, u\}, \{a, v\}, \{b, u\}$, $\{b, v\}$ black;
        color $\{b, c\}, \{c, a\}$ brown;
        return $\mathcal{B}(G - \{a, b\}, \eta \wedge \alpha(G[a, b, x, y]))$;
    else if *there is a separating 4-cycle* $C = (a, b, c, d)$ *with* $G - C = \{G_{in}, G_{out}\}$ **then**
        extend the coloring by brown edges $\{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}$;
        if $\{a, c\} \in G_{in}$ **then** $e = \{b, d\}$ **else** $e = \{a, c\}$;
        if $\{a, c\} \in G_{out}$ **then** $f = \{b, d\}$ **else** $f = \{a, c\}$;
        return merge($\mathcal{B}(G_{in} + C + e, \eta_{in}), \mathcal{B}(G_{out} + C + f, \eta_{out})$);
    else if *there is a separating triangle* $C = (a, b, c)$ **then**
        return $\mathcal{B}(tri(G, C, \eta))$;
    else if *there is a separating quadruple* $C = (a, b, c, d)$ **then**
        return $\mathcal{B}(quad(G, C, \eta))$;
    else if *there is a* K_5 **then return** (G, false) and stop ;
    else $MC_4(G,\ \eta)$;
Final-Check(G, η);

Algorithm 2: Algorithm MC_4

Input: A 5-connected graph G with a partial edge coloring and a boolean formula η.

Output: A subgraph of G, an edge coloring, and η.

if *the detected* K_4 *is a completely kite-covered tetrahedron* T **then**
    foreach *edge* e *of* T **do**
        color $e = \{a, c\}$ red and its crossing edge $f = \{b, d\}$ blue and extend the coloring by black edges $\{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}$;
        set $\eta = \eta \wedge \alpha(\kappa)$;
    collect the red edges into a set F;
    return $(G - F, \eta)$;
else if *the detected* K_4 *subgraph is an SC graph* **then**
    foreach *kite* κ *of the SC graph with a pair of crossing edges* e, f **do**
        color e red and f blue and color the remaining edges black;
        set $\eta = \eta \wedge \alpha(\kappa)$;
    collect the red edges into a set F;
    return $(G - F, \eta)$;
else
    color the crossing edges of the detected kite κ red and blue and color the other edges of κ black;
    set $\eta = \eta \wedge \alpha(\kappa)$;
    return $(G - e, \eta)$ where e is the red edge;

given edge coloring and it returns an embedding of a planar subgraph. Otherwise it fails. A description is given in Algorithm 3, which also computes an edge coloring and a boolean formula.

The correctness of algorithm $\mathcal{B}$ uses the following properties of Algorithm $\mathcal{A}$, as proved in [16].

Lemma 4 *Let $\mathcal{E}(G)$ be any triangulated 1-planar embedding of G. Then algorithm $\mathcal{B}$ succeeds on G and returns an embedding of a triangulated planar spanning subgraph inherited from $\mathcal{E}(G)$, an edge coloring, and a boolean formula η such that black, grey and brown edges are non-crossed in $\mathcal{E}(G)$, red, blue, and orange edges are crossed in $\mathcal{E}(G)$, there are pairs of red and blue edges that cross, and each orange edge crosses a cyan one, except in subgraphs that are simple K_5. Moreover, one edge from each pair of crossing edges is red, or orange. Each black edge is part of a kite, each grey edge is part of a possible kite, and each brown edge is not related to a kite at all.*

Proof: By the assumptions, G is a triangulated 1-planar graph and algorithm $\mathcal{A}$ succeeds on G. Since $\mathcal{B}$ extends $\mathcal{A}$, it also succeeds, since MC_5 does not apply, as shown in [9]. Then $\mathcal{B}$ provides the edge coloring as stated. All K_4 subgraphs are scanned and their embedding is classified as a kite, as a possible kite at a separating edge or a small graph, as a planar tetrahedron (after a separating 3-cycle), or as a tetrahedron with crossed edges in a kite-covered tetrahedron or an SC graph. One edge of each detected kite is colored red or orange, and the non-crossed edges are colored black, grey or brown according to the coloring scheme. There is an exception for simple K_5 subgraphs since the pair of crossing edges is unclear. □

5 Specialization to IC- and NIC-Planarity

For the test of IC- and NIC-planarity we can specialize MC_4 and final-check. For example, a completely kite-covered K_4 is not IC-planar and the SC-graph is neither IC- nor NIC-planar. If such a subgraph is encountered, then the recognition algorithm for IC-planarity (NIC-planarity) fails and returns false and stops.

It remains to test whether the 1-planar embedding computed by algorithm $\mathcal{B}$ is IC-planar (NIC-planar) or can be transformed into such an embedding. The main steps of algorithm $\mathcal{B}$ uniquely determine whether an edge is crossed or non-crossed and express this property by the edge coloring. There is an ambiguity at separating edges and small graphs, for which it suffices to consider only embeddings in which vertices or edges of the outer face must be a part of a kite. These cases are expressed by a disjunction in the boolean formula.

Simple K_5 are a particular case. They result from separating 3-cycles. Let the 3-cycle describe the outer face and consist of vertices a, b, c and edges e, f, g and let x and y be the inner vertices, see Fig. 3. It admits three 1-planar embeddings up to isomorphism with x and y interchanged, such that each edge of the outer face is part of a kite, as illustrated in Fig. 3. The inner edge $\{x, y\}$ is part of each of the three possible kites and each edge $\{u, v\}$ between an outer and an inner vertex can be crossed. The ambiguity in the embeddings is due to six separating edges between each outer and each inner vertex. For example, $\{a, y\}$ may cross $\{b, x\}$ or $\{c, x\}$. Hence, separating edges are the deeper reason for an ambiguity in 1-planar (IC-planar, NIC-planar) embeddings of triangulated 1-planar graphs. They are specialized to *extended* 4^+*-wheel graphs* $X4W$, which are embeddings of $K_5 - e$ with an outer 4-cycle, as shown in Fig. 5. The coloring scheme is exempted for the edges of simple K_5. Fortunately, this deficit is captured by the boolean formulas. The set of embeddings is expressed by $\alpha = (a_{\kappa_1} \wedge b_{\kappa_1}) \vee (a_{\kappa_2} \wedge c_{\kappa_2}) \vee (b_{\kappa_3} \wedge c_{\kappa_3})$ in the IC-planar case and by $\alpha = (e_{\kappa_1} \vee f_{\kappa_2} \vee g_{\kappa_3})$ in the NIC-planar case.

If two simple K_5 are adjacent, as shown in Fig. 15(b), with vertices a, b, c and b, c, d in the outer face, then there are two IC-planar embeddings represented by the formula $(a_{\kappa_1} \wedge b_{\kappa_1} \wedge c_{\kappa_2} \wedge d_{\kappa_2}) \vee (a_{\kappa_3} \wedge c_{\kappa_3} \wedge b_{\kappa_4} \wedge d_{\kappa_4})$, where $\kappa_1, \ldots, \kappa_4$ are the possible kites of the two K_5. These formulas simplify to $(a \wedge b \wedge c \wedge d)$, since each outer vertex is in a kite. A NIC-planar embedding that uses the diagonal $\{b, c\}$ for one kite and one outer edge for the second kite leads to the formula $(e_{\kappa_1} \vee f_{\kappa_2} \vee g_{\kappa_3} \vee h_{\kappa_4})$, where e, f, g, h are the edges of the outer face and $\kappa_1, \ldots, \kappa_4$ are possible kites.

Algorithm 3: Algorithm Final-Check

Input: A 3-connected graph G with a partial edge coloring and a boolean formula η.

Output: An embedded planar subgraph of G, an edge coloring and η.

- **if** *G is a triangulated planar graph* **then**
  - extend the coloring by brown edges;
  - **return** $(\mathcal{E}(G), \eta)$;
- **else**
  - **if** $|G| \leq 8$ **then**
    - **if** *G has an IC-planar (NIC-planar) embedding extending the partial edge coloring γ* **then**
      - **if** *there is $\mathcal{E}(G)$ without vertices (edges) in the outer face in a kite* **then**
        - set $\beta =$ true
      - **else**
        - set $\beta =$ false;
        - **foreach** *IC-planar (NIC-planar) embedding $\mathcal{E}(G)$ extending γ* **do**
          - express $\mathcal{E}(G)$ by a boolean formula α using only variables for the vertices (edges) in the outer face of $\mathcal{E}(G)$ (with the added chord ignored);
          - set $\beta = \beta \vee \alpha$;
      - set $\eta = \eta \wedge \beta$;
      - color the edges of G according to the coloring scheme;
      - choose any IC-planar (NIC-planar) embedding $\mathcal{E}(G)$ with a set F of red or orange edges that are crossed in $\mathcal{E}(G)$;
      - **return** $(\mathcal{E}(G - F), \eta)$;
  - **return** $(G,$ false$)$ and stop

The extended version of algorithm final-check in $\mathcal{B}_{IC}$ ($\mathcal{B}_{NIC}$) computes all IC-planar (NIC-planar) embeddings of the small graph G of size at most eight, that extend the given partial coloring and that inevitably use a vertex (edge) of the outer face as part of a kite. Each embedding is expressed by a boolean formula using only variables for the vertices (edges) in the outer face. These formulas are combined by a disjunction to express the set of all IC-planar (NIC-planar) embeddings.

The boolean formula η collects the clause $\alpha(\kappa)$ of all possible kites κ, but it does not express IC- and NIC-planarity and a mutual exclusion of two kites with a common vertex and edge, respectively. Therefore, we extend η to η^+, and transform η^+ into η^* for an efficient evaluation.

Definition 5 *The* IC-extension *(*NIC-extension*) η^+ of a boolean formula η is* *obtained by adding a clause $(\neg x_\kappa \vee \neg x_{\kappa'})$ if there is a vertex (edge) x with two variables x_κ and $x_{\kappa'}$ and $\kappa \neq \kappa'$ in clauses of η.*

Let $\mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$ be the algorithms obtained from algorithm $\mathcal{B}$ by the versions for IC-planar and NIC-planar graphs, respectively. $\mathcal{B}_{IC}$ returns false and stops if $\mathcal{B}$ encounters a separating triangle, if MC_4 encounters a completely kite-covered K_4 or an SC-graph or two kites that share a vertex, and it uses algorithm *Final-Check* for the final test. Accordingly, $\mathcal{B}_{NIC}$ returns false and stops if MC_4 encounters an SC-graph or if MC_4 detects two kites that share an edge, and it uses algorithm *Final-Check* for the final test.

Theorem 2 *A graph G is triangulated IC-planar (NIC-planar) if and only if Algorithm $\mathcal{B}_{IC}$ ($\mathcal{B}_{NIC}$) succeeds and returns a boolean formula η whose IC-extension (NIC-extension) η^+ is satisfiable.*

Proof: If algorithm $\mathcal{B}_{IC}$ ($\mathcal{B}_{NIC}$) succeeds, then G is triangulated 1-planar by Lemma 4, since $\mathcal{B}_{IC}$ ($\mathcal{B}_{NIC}$) specializes algorithm $\mathcal{B}$, and an embedding $\mathcal{E}(S)$ of the planar subgraph S of G without the removed red and orange edges is returned. Construct an embedding $\mathcal{E}(G)$ from $\mathcal{E}(S)$ by re-inserting the removed edges and taking a satisfying truth assignment of η^+ into account.

First, suppose that G is not small, i.e., G is not a subgraph of size at most eight that is a component with a separating 3- or 4-cycle. Each red edge has a unique blue one for a crossing and the subgraph induced by the endvertices of the crossing edges is embedded as a kite κ, whose further edges are uncrossed and are colored black. Then $\alpha(\kappa)$ is a clause of η and of η^+, and each variable x_κ in $\alpha(\kappa)$ is assigned the value true. If e is an orange edge, then it is a separating edge e and has the choice among its separating crossable edges $f_1, \ldots, f_k$ for some $k \geq 2$ for a crossing. All possible kites κ_i with a crossing of e and some f_i are described by $\alpha(\kappa_1) \vee \ldots \vee \alpha(\kappa_k)$ in η and η^+. If η^+ is satisfied, then $\alpha(\kappa_i)$ is satisfied for at least one i with $i = 1, \ldots, k$, and any such i is chosen so that e crosses f_i in $\mathcal{E}(G)$.

Next, consider small subgraphs H whose outer face is a 3-cycle or 4-cycle. Then H has an IC-planar (NIC-planar) embedding that extends the edge coloring, since otherwise the algorithms would fail. The coloring of the edges of H may not agree with the coloring scheme, e.g., if H is a simple K_5, but this does not matter. Again, consider an IC-planar (NIC-planar) embedding of H such that a vertex (edge) x in the outer face of H is part of a kite if the associate variable x_κ is assigned the value true.

By construction and Lemma 4, $\mathcal{E}(G)$ is a 1-planar embedding and it is IC-planar (NIC-planar), since the truth assignment guarantees that each variable x_κ of a kite κ of $\mathcal{E}(G)$ is assigned the value true by a truth assignment that satisfies η^+ and the IC-extension (NIC-extension) excludes a use of the corresponding vertex (edge) in another kite. Hence, $\mathcal{E}(G)$ is IC-planar (NIC-planar).

Conversely, if G is a triangulated IC-planar (NIC-planar) graph, then algorithm $\mathcal{B}_{IC}$ ($\mathcal{B}_{NIC}$) succeeds and returns a boolean formula η describing all 1-planar embeddings of G, except for small graphs for which only the use of vertices (edges) of the outer face is described. The algorithms check all K_4 subgraphs. For every possible kite κ there is a subformula $\alpha(\kappa)$ in η that describes κ. Consider an IC-planar (NIC-planar) embedding $\mathcal{E}(G)$. If a vertex (edge) x is part of a kite κ in $\mathcal{E}(G)$, then assign x_κ the value true and assign false, otherwise. Then η is satisfiable, since κ and the variables x_κ were taken into account by the algorithms. By IC-planarity (NIC-planarity) each vertex (edge) is part of at most one kite so that also the IC-extension (NIC-extension) η^+ is satisfied. □

Algorithm $\mathcal{A}$ runs in cubic time [16], and so do $\mathcal{B}, \mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$. It thus remains to solve the satisfiability problem of η^+, which is investigated in the next section and takes $O(n^2)$ time if the given graph is IC-planar or NIC-planar and K_5-free or $X4W$-free. Thus NIC-planar graphs shall not contain either simple K_5 or extended wheel graphs, $X4W$ for short, which are made precise in Definition 7. $X4W$-graphs comprise separating edges and 4^+-wheel graphs so that pairs of edges may be in a kite. These subgraphs can be recognized by $\mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$ in triangulated 1-planar graphs.

Triangulated 1-planar embeddings are maximized in the sense that no non-crossed edge can be added without violating 1-planarity. However, $K_5 - e$ admits a 1-planar embedding with a pair of crossing edges so that edge e can be added non-crossed. We call a graph *maximal* (*planar-maximal*) 1-planar if no edge e can be added to G so that $G + e$ admits a 1-planar drawing (in which e is non-crossed). If $G + e$ violates the upper bound of the number of edges of 1-planar graphs, then G is *maximum* or *densest* and G is called *optimal* if the number of edges exactly meets the upper bound of $4n - 8$. Similar notions apply to planar, IC-planar, and NIC-planar graphs. Clearly, every triangulated planar graph is optimal, and there are (planar) maximal IC-planar, NIC-planar [5] and 1-planar graphs [12] that are not optimal.

We can test maximality by exhaustive search on all possible edges e, such that graph G is 1-planar (IC-planar, NIC-planar) and $G + e$ is not. In the planar-maximal case, edge e is colored brown, and therefore is embedded non-crossed.

Corollary 1 *For a graph G it takes $\mathcal{O}(n^5)$ time to test whether G is planar-maximal or maximal 1-planar IC-planar (K_5-free or $X4W$-free NIC-planar).*

There are special linear time algorithms for optimal 1-planar graphs [10] and optimal NIC-planar graphs [5], which use the particular structure of optimal graphs. Note that there are optimal IC-planar (NIC-planar) graphs only for values $n = 4k$ ($n = 5k + 2$) and $k \geq 2$.

Corollary 2 *For a graph G it takes $\mathcal{O}(n^3)$ time to test whether G is maximum (optimal) IC-planar.*

Proof: Clearly, a maximum (or densest) graph is triangulated. It has $\lfloor 13/4\, n - 6 \rfloor$ edges if it IC-planar. □

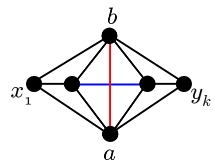


Figure 8: A C-unambiguous IC-planar (NIC-planar) graph

6 IC- and NIC-planarity of 1-planar Embeddings

The algorithms $\mathcal{B}$ and similarly $\mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$ compute the "planar skeleton" of a given triangulated 1-planar graph consisting of the black, brown and grey edges that are embedded non-crossed. Moreover, they determine edges that cross inevitably and color them red, blue and orange, respectively. There is an ambiguity in the embeddings if there is a separating edge, which may choose among its separating crossing edges for a crossing, and at small graphs that result from separating 3- or 4-cycles. The ambiguity is expressed by a disjunction in the computed boolean formula η.

Definition 6 *An IC-planar (NIC-planar) embedding $\mathcal{E}(H)$ of a graph H with cycle C is C-*minimal IC-planar *(*NIC-planar*) if C describes the outer face of $\mathcal{E}(H)$ and the set S of vertices (edges) of C that is part of a kite is minimal w.r.t such embeddings.*

A (small) graph H of size at most eight with a 3- or 4-cycle C is called IC-ambiguous *(*NIC-ambiguous*) if it has at least two C-minimal IC-planar (NIC-planar) embeddings.*

For example, a simple K_5 with a fixed outer face with vertices a, b, c and edges e, f, g is IC-ambiguous (NIC-ambiguous). There are three C-minimal embeddings with a pair of vertices (edge) of C. The set of embeddings is expressed by $\eta = (a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$ for IC-planar graphs and by $\eta = (e \vee f \vee g)$ for NIC-planar graphs. Also, the 4^+-wheel graph is IC-ambiguous (NIC-ambiguous).

A small graph is unambiguous if it admits an embedding so that some vertices (edges) of the outer face are not needed for a kite, as shown in Fig. 8. If no vertex (edge) of C is part of a kite, then H is treated as planar and $\eta = \mathsf{true}$ is returned by $\mathcal{B}_{IC}$ ($\mathcal{B}_{NIC}$).

6.1 Evaluate IC-planar Formulas

In the IC-planar case, the computed boolean formula is in 2CNF, except if there is an IC-ambiguous graph or a separating edge. For IC-ambiguous graphs there is a transformation into 2CNF and separating edges are treated by a series of 2SAT problems.

Lemma 5 *If G is an IC-planar graph without separating edges, then the IC-extension η^+ of the formula η computed by algorithm $\mathcal{B}_{IC}$ is equivalent to a 2SAT formula η^*.*

Proof: The boolean formula $\alpha(\kappa) = (a_\kappa \wedge b_\kappa \wedge c_\kappa \wedge d_\kappa)$ of a kite $\kappa = G[a, b, c, d]$ is added to η at a separating triple, a separating quadruple, and a kite in MC_4. If a small graph H is IC-unambiguous, then a conjunction of variables or even true is added to η. Two subexpressions are combined by a conjunction at separating 3- and 4-cycles. Also the IC-extensions are 2SAT formulas.

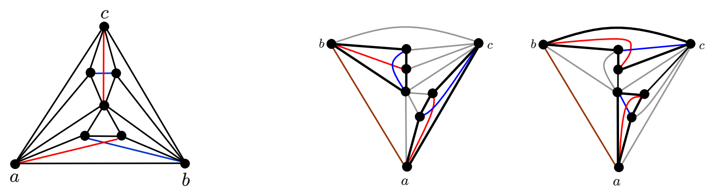
It thus remains to show that the IC-extension of a formula η for an IC-ambiguous graph H with outer face C is equivalent to a 2SAT formula, and is replaced by the 2SAT formula for further computations. Every IC-planar embedding contains at most one kite if $|H| \leq 7$. Also, we can assume that edges in $H - C$ are uncolored; otherwise the set of possible IC-planar embeddings is further restricted.

First, suppose that $C = (a, b, c)$ is a 3-cycle. If $|H| = 5$, then H is a K_5 so that $\eta = (a_{\kappa_1} \wedge b_{\kappa_1}) \vee (a_{\kappa_2} \wedge c_{\kappa_2}) \vee (b_{\kappa_3} \wedge c_{\kappa_3})$. Together with the IC-extension, the expression is equivalent to the 2SAT formula $(a \vee b) \wedge (a \vee c) \wedge (b \vee c)$ and is replaced by this formula. If H has 6, 7 or 8 vertices and every C-minimal IC-planar embedding has one kite that includes a, b or c, then by IC-ambiguity, H contains a K_5 with two or three vertices from a, b, c, where a third vertex occurs in every possible kite. Then $(x \vee y) \wedge z$ or $x \vee y$ describe the IC-planar embeddings with variables x, y, z for vertices of C. If there are two kites κ and κ', then H has eight vertices and each vertex must be in a kite, so that $\eta = (a \wedge b \wedge c)$ describes the embeddings. An IC-ambiguous graph is shown in Fig. 9(b).

Accordingly, if $C = (a, b, c, d)$ is a 4-cycle, then H is a 4^+-wheel graph if $|H| = 5$ so that $\eta = (a_\kappa \wedge b_\kappa \wedge c_\kappa) \vee (a_{\kappa'} \wedge d_{\kappa'} \wedge c_{\kappa'})$, which after the IC-expansion is equivalent to the 2SAT formula $a \wedge c \wedge (b \vee d)$. The same expression occurs if $\{a, c\}$ is a separating edge with crossable edges $\{b, u\}$ and $\{d, v\}$ for some inner vertices u and v and H has 7 or 8 vertices, see Fig. 12(a). If every embedding of H has two kites, as shown in Fig. 10, then $|H| = 8$ and each vertex of H is in a kite. After the IC-extension the formulas are equivalent to $\eta = (a \wedge b \wedge c \wedge d)$. Otherwise, if every C-minimal IC-planar embedding has one kite κ with vertices of C, then H contains a K_5 or a 4^+-wheel graph with two or three vertices of C so that $(x \vee y) \wedge (x \vee z) \wedge (y \vee z)$, $(x \vee y)$ or $x \wedge (y \vee z)$ describe the embeddings after the IC-extension with variables x, y, z for the vertices of C that may be in a kite, as shown in Figs. 11 and 12(b) □

The satisfiability problem for η^+ can be reduced to a series of 2SAT satisfiability problems, which altogether can be solved in linear time in the length of η^+. This technique was used in [11].

Lemma 6 *There is a linear time algorithm to test the satisfiability of the IC-extension η^+ of a triangulated IC-planar graph.*



(a) (b)

Figure 9: IC-planar embeddings of small graphs with a 3-cycle. (a) is a unique IC-planar embedding and is IC-unambiguous and (b) is IC-ambiguous.

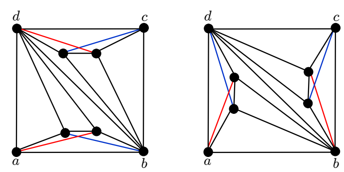
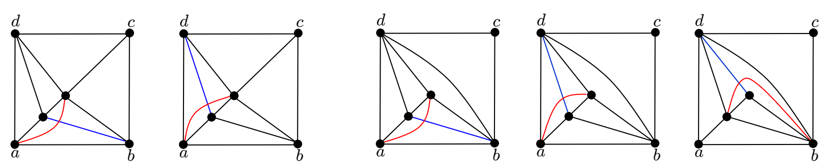
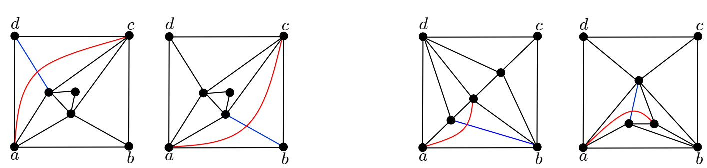


Figure 10: IC-ambiguous graphs with a 4-cycle and two K_5



(a) (b)

Figure 11: IC-ambiguous graphs with 6 vertices



(a) (b)

Figure 12: IC-ambiguous graphs with 7 vertices

Proof: Consider the recursive construction of the boolean formula η by algorithm $\mathcal{B}_{IC}$. All subexpressions are in 2SAT form or are replaced by an equivalent 2SAT formula as shown in Lemma 5, except if there is a separating edge.

Let $\{a, b\}$ be a separating edge with $k \geq 2$ separating crossable edges $f_i = \{x_i, y_i\}$ such that $G - \{a, b\} - \{f_i \,|\, i = 1, \ldots, k\}$ partitions into k components G_i with y_i in G_i for $i = 1, \ldots, k$, see Fig. 4. Let $(x_1, y_1, \ldots, x_k, y_k)$ be the cyclic ordering at a according to the rotation system, which is obtained from the embedding returned by $\mathcal{B}_{IC}$. Then $x_{i+1} \in G_i$ for $i = 1, \ldots, k-1$ and x_1 in G_k. There is a separating 4-cycle $C_i = (a, y_i, b, x_{i+1})$ with $x_{k+1} = x_1$ separating G_i from the other components. We call G_k the outer component and G_i with $i = 1, \ldots, k-1$ inner components. In fact, any component can take the role of the outer component.

We proceed by induction on the depth of nesting of separating edges. Consider a separating edge $\{a, b\}$ on the least level such that there are no separating edges in inner components. The inner components are not necessarily subgraphs of G but may have been modified as described below.

The boolean formula $\sigma(a, b, \mathcal{C}[a, b])$ describing a separating edge can be transformed into $(a \wedge b) \wedge ((x_1 \wedge y_1) \vee \ldots, \vee (x_k \wedge y_k))$ with boolean variables a, b, x_i, y_i for $i = 1, \ldots, k$ associated with the vertices and possible kites κ_i. We wish to use a pair of vertices (x_i, y_i) for $i = 2, \ldots, k-1$ between two inner components together with a and b for the kite so that the vertices x_1 and y_k of the outer component G_k are "free" for a kite in the outer component G_k.

For $i = 2, \ldots, k-1$, edges $\{a, b\}$ and f_i can cross according to IC-planarity if and only if the subgraph H_i consisting of G_{i-1}, a kite κ_i with vertices a, y_i, b, x_{i+1}, and G_i admits an IC-planar embedding, where we assume that subgraphs G_i also include the vertices a and b and the incident edges to the components. Consider the boolean formula η_i for H_i, which is obtained from the boolean formula η for G by the restriction to (variables for) vertices in H_i and the substitution of separating edges and the corresponding formulas in the inner components H_i. By induction, H_i has no separating edges, such that the IC- extension of η_i is equivalent to a 2SAT formula ξ_i^+. The existence of kite κ_i in H_i is expressed by $\alpha(\kappa_i) = (a_{\kappa_i} \wedge b_{\kappa_i} \wedge y_{i,\kappa_i} \wedge x_{i+1,\kappa_i})$.

The satisfiability of ξ_i^+ can be checked in linear time in the length of ξ_i^+. If ξ_i^+ is satisfiable, then we replace the subexpression $\sigma(a, b, \mathcal{C}[a, b])$ of η^+ by $(a \wedge b)$. Thereby all variables corresponding to vertices from the inner components are removed. This means a replacement of the inner components $G_1, \ldots, G_{k-1}$ by a subgraph as shown in Fig. 8.

If neither of the ξ_i^+ for $i = 2, \ldots, k-1$ is satisfiable, in particular if $k = 2$ and there is just one inner component, then either of the extreme separating crossable edges f_1 or f_k must be used in a kite and either x_1 or y_k is part of a kite. Then we replace $\sigma(a, b, \mathcal{C}[a, b])$ by $(a \wedge b) \wedge (x_1 \vee y_k)$ which corresponds to a replacement of the inner components by a 4^+-wheel graph. Thereby the separating edge is removed, and we proceed with the modified outer component in which the inner components have been replaced by the 4^+-wheel graph.

The subexpression of each inner component G_i is evaluated at most twice, for H_{i-1} and for H_i, and subexpressions for distinct components are distinct.

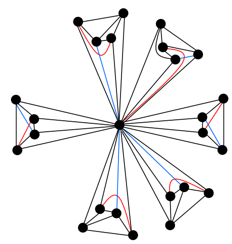


Figure 13: A graph with many K_5 subgraphs inducing many IC-extensions

Each subexpression can be evaluated in linear time so that the satisfiability test of η^+ takes linear time in its length. □

A 1-planar graph of size n has at most $n-2$ kites, and the bound is achieved by optimal 1-planar graphs with $4n - 8$ edges [7, 32]. Hence, the boolean formula η has length $O(n)$. However, the IC-extension may add up to $O(n^2)$ subexpressions of the form $(\neg x_\kappa \vee \neg x_{\kappa'})$, for example, if x is the center of a star of K_5, as illustrated in Fig. 13.

In consequence, the satisfiability check of η^+ takes linear time in the length of the formula and at most quadratic time in the size of the input graph. This is less than the cubic running time of algorithm $\mathcal{B}_{IC}$.

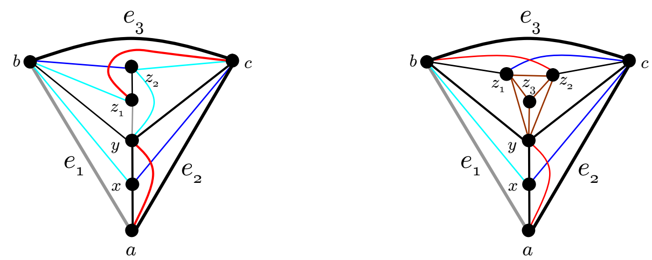
In summary, we obtain:

Theorem 3 *Triangulated IC-planar graphs can be recognized in cubic time.*

6.2 Evaluate NIC-planar formulas

Also in the NIC-planar case, the boolean formula η has a special form, however, a fast satisfiability test is not known. The formula is a conjunction of terms of the form $\beta = (a \wedge b) \vee \ldots \vee (c \wedge d)$ and $\gamma = (a \vee b \vee c)$ or subterms thereof. Subterms β describe the set of NIC-planar embeddings of separating edges and in particular of small graphs containing a 4^+-wheel graph, whereas the 3SAT formula γ describes the set of NIC-planar embeddings of simple K_5. As before we can solve the case of separating edges by a series of subproblems without separating edges in inner components. Subformulas β with only two conjunctions can be transformed into an equivalent 2SAT formula $(a \vee c) \wedge (a \vee d) \wedge (b \vee c) \wedge (b \vee d)$. However, 3SAT formulas from K_5 subgraphs remain. In addition, there are at most two boolean variables for each edge so that the NIC-extension η^+ can be transformed into a boolean formula η^* in which each variable and its negation occur exactly once, so that η^+ is satisfiable if and only if η^* is satisfiable.

Lemma 7 *If a graph H is NIC-ambiguous and the outer face is a 3-cycle C, then H is K_5 and the set of NIC-planar embeddings is expressed by a clause $\alpha = (e_1 \vee e_2 \vee e_3)$ or H contains K_5 and $\alpha = (e_1 \vee e_2)$ or $\alpha = ((e_1 \vee e_2) \wedge e_3)$.*



(a) (b)

Figure 14: Embeddings of NIC-ambiguous graphs with a 3-cycle

Proof: Let $\mathcal{E}_i(H)$ be two minimal NIC-planar embeddings of H such that C consists of edges e_1, e_2, e_3 and e_i is in a kite in $\mathcal{E}_i(H)$ for $i = 1, 2$. Let $C = (a, b, c)$ and suppose that $e_1 = \{a, b\}$ and $e_2 = \{a, c\}$. All other cases are symmetric. If H has five vertices, then there is an inner vertex y such that $\{a, y\}$ is a separating edge that must cross one of $\{x, b\}$ or $\{x, c\}$ for an inner vertex x. Now H is a K_5 so that $\alpha = (e_1 \vee e_2 \vee e_3)$.

If there are further inner vertices z_j with $1 \leq j \leq 3$, then they must be embedded in the cycle (b, c, y), if y is a neighbor of some z_j and $\{a, y\}$ is a separating edge, see Fig. 14. Then x is not a neighbor of any z_j, since $\{a, y\}$ is a separating edge. Otherwise, x and y change roles. Since H is NIC-ambiguous, the edges incident to z_j can be embedded non-crossed, for example if $|H| = 6$, so that $\alpha = (e_1 \vee e_2)$. Otherwise, $H[b, c, y, z_1, z_2]$ is a K_5 which may be embedded with $\{b, c\}$ in a kite if $|H| = 7$, as shown in Fig. 14(a), but there is no need for it, since instead $\{b, y\}$ or $\{c, y\}$ can be used in the second kite. The shown embedding with $\{b, c\}$ is not C-minimal. Thus the embeddings are described by $(e_1 \vee e_2)$. If H has eight vertices, then (b, c, y) is a separating 3-cycle of H. Then there is a fixed kite $H[b, c, z_1, z_2]$ that includes $\{b, c\}$, as displayed in Fig. 14(b), so that $((e_1 \vee e_2) \wedge e_3)$ describes all NIC-planar embeddings of H. Otherwise, $H[b, c, z_1, z_2, z_3]$ can be drawn planar or contains a 4^+-wheel graph or a K_5, so that $(e_1 \vee e_2)$ describes the NIC-planar embeddings of H. □

Similarly, there are four cases if there is a separating 4-cycle.

Lemma 8 *If a NIC-ambiguous graph H has an outer 4-cycle C, then H consists of two K_5, as shown in Figs. 15(a) and 15(b), if and only if $\alpha = (e_1 \vee e_2 \vee e_3 \vee e_4)$ is a 4-clause. Otherwise, α is a single disjunction of terms of the form $(e \wedge f)$ for variables e and f, i.e, $\alpha = (e_1 \wedge e_2) \vee (e_3 \wedge e_4)$, or a subterm thereof.*

Proof: Suppose that $C = (a, b, c, d)$ and let $e_1 = \{a, b\}$, $e_2 = \{b, c\}$, $e_3 = \{c, d\}$ and $e_4 = \{a, d\}$ represent the edges and their corresponding boolean variables. All other cases are symmetric. Let u, v, x, y be the remaining inner vertices of H.

If G has five vertices, then H is a 4^+-wheel graph with chord $\{b, d\}$, as shown in Fig. 5, and the set of embeddings is described by $\alpha = (e_1 \wedge e_2) \vee (e_3 \wedge e_4)$. If there is a chord $\{a, c\}$, then let $\alpha = (e_1 \wedge e_4) \vee (e_2 \wedge e_3)$. Similarly, if H has 7 or 8 vertices and a separating edge $\{a, c\}$ as shown in Fig. 12(a), we have $\alpha = (e_1 \wedge e_2) \vee (e_3 \wedge e_4)$.

If H contains two K_5, then is has eight vertices and $\alpha = (e_1 \vee e_2 \vee e_3 \vee e_4)$, see Fig. 15(a), since one K_5 is embedded at $\{b, d\}$.

Next, we show that there is no other NIC-ambiguous graph H with a 4-cycle such that $\alpha = (e \vee f \vee g)$ or $\alpha = e \vee f \vee (g \wedge h)$ describes the set of NIC-planar embeddings of H, where e, f, g and h are (variables for) edges of C. For NIC-ambiguity there must be a K_5 or a 4^+-wheel graph, say $H[a, b, d, y, x]$ with chord $\{a, y\}$. Then $C' = (b, c, d, y)$ is a separating 4-cycle. If $\{b, d\}$ exists, then either $H[b, c, d, u, v]$ is another K_5 so that H contains two K_5 and α is a 4-clause as stated above, or $\{b, d\}$ is used for the first kite and H is not C-minimal NIC-planar. Otherwise, if e_2 may be in a kite and H has seven vertices, then $H[b, c, d, y, u]$ is a 4^+-wheel graph with chord $\{y, c\}$, see Fig. 16(a), and $\alpha = (e_1 \wedge e_3) \vee (e_2 \vee e_4)$. If H has eight vertices, a term $\alpha = (e_1 \vee e_4 \vee \beta)$ implies that edges $\{b, y\}$ and $\{d, y\}$ are not part of another kite. Then edges e_2 and e_3 may be in another kite if $H[b, c, d, u, v]$ is a 4^+-wheel graph. Depending on the chord, either $\alpha = ((e_1 \vee e_4) \wedge f)$ with $f = e_3$ or $f = e_4$ or $\alpha = (e_1 \vee e_4)$ describe the NIC-planar embeddings, see Fig. 16(b) □

Note that there are NIC-ambiguous graphs with a 4-cycle so that either two pairs of adjacent edges or two pairs of opposite edges are part of a kite, i.e., $\alpha = (e_1 \wedge e_2) \vee (e_3 \wedge e_4)$ for a 4^+-wheel graph, or $\alpha = (e_1 \wedge e_3) \vee (e_2 \wedge e_4)$ from the double 4^+-wheel graph in Fig. 16(a).

The boolean formula η obtained by algorithm $\mathcal{B}_{NIC}$ has another special feature.

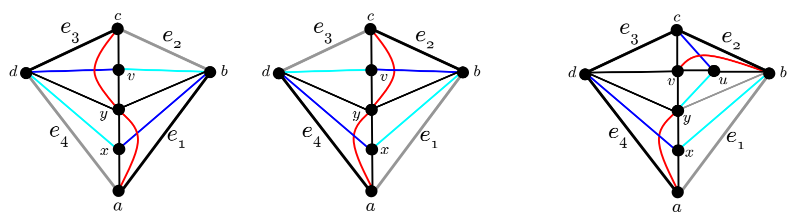
Lemma 9 *For every edge e with an occurrence of a variable e_κ in η there are at most two variables e_κ and $e_{\kappa'}$ in η.*

Proof: First, observe that each variable e_κ occurs once in η, since algorithms $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{B}_{NIC}$ check each K_4 exactly once and $\mathcal{B}_{NIC}$ introduces a boolean variable e_κ if edge e is a non-crossed edge of a possible kite κ, which is a unique event.

We claim that there is at most one possible kite on either side of an edge e in a NIC-planar embedding. For small graphs there are formulas of the form $(e_1 \vee e_2 \vee e_3 \vee e_4)$ or $(e_1 \wedge e_2) \vee (e_3 \wedge e_4)$ or subterms thereof, where $e_1, \ldots, e_4$ are variables for edges of a separating 3- or 4-cycle. Each major step except for a separating 3- or 4-cycle introduces exactly one variable e_κ for each edge that may be part of a kite in an embedding of G. Finally, if e is an edge of a separating 3- or 4-cycle, then it is associated with a variable e_κ if it is colored black or grey and embedded non-crossed. If e is colored blue, then it is a crossing edge of a kite and there is no variable e_κ. Suppose that $G[a, b, u, v]$ and $G[a, b, x, y]$ are K_4 that are possibly embedded as a kite and include edge $e = \{a, b\}$ of C, so that u, v, x and y are in the same component of $G - C$.

(a) (b)

Figure 15: NIC-planar embeddings of a small graph with 8 vertices, a planar 4-cycle and two K_5



(a) (b)

Figure 16: NIC-planar embeddings of NIC-ambiguous graphs with 8 vertices

Since G is 3-connected u, v, x, y cannot be distinct. If two vertices coincide, say $v = x$, then $G[a, b, u, v, y]$ is (a subgraph of) K_5 and there is one variable e_κ for e. Hence, the claim is true. □

The extreme cases from Lemmas 7 and 8 lead to the following specialization.

Definition 7 *A NIC-planar graph G is K_5-free if it does not contain a simple K_5 as shown in Fig. 3.*

Graph G is $X4W$-free (extended 4-wheel graph) if it does not have separating edges and there are no small subgraphs H as follows: there is a separating 4-cycle $G[a, b, c, d]$, which is the outer face of an embedding of H, and H is a 4^+-wheel graph (see Fig. 5=, a double 4^+-wheel graph *(see Fig. 16(a)), or has a separating edge $\{a, c\}$ (see, e.g., Fig. 12(a)).*

Note that a K_5-free graph may contain K_5 as a subgraph. Then K_5 is embedded so that at least three of its edges are crossed. Such embeddings are recognized by the algorithms using separating triples or separating triangles. Similarly, 4^+-wheel graphs may appear in small subgraphs such that the edges of the outer face are not needed in pairs for a kite and there is a choice among the pairs of edges.

Lemma 10 *There is a quadratic time algorithm that checks whether a triangulated NIC-planar graph is K_5-free or $X4W$-free.*

Proof: All 3-cycles (4-cycles) of a 1-planar graph can be computed and listed in linear time using the algorithm by Chiba and Nishizeki [17]. For each cycle C it takes linear time to test whether it is separating. Now all simple K_5, i.e., K_5 with a separating 3-cycle, can be found in $O(n^2)$ time. It takes $O(n^2)$ time to test whether a triangulated 1-planar graph has a separating edge [16] and linear time whether it contains a 4-cycle together with a forbidden graphs, such as a (double) 4^+-wheel graph. □

Lemma 11 *The satisfiability of η^+ can be reduced to the satisfiability of a boolean formula η^* that is a conjunction of terms of the form $(a \wedge b) \vee (c \wedge d)$ or $(a \vee b \vee c \vee d)$ and subterms thereof, such that each variable x and its negation $\neg x$ occur exactly once in η^*.*

Proof: Consider the formula η that is obtained by algorithm $\mathcal{B}_{NIC}$. It consists of a conjunction of terms of the described form. We simplify η and use its NIC-extension η^+ while preserving satisfiability.

Delete e_κ from η if there is a single variable e_κ for some edge e. Then there is no NIC-extension with e_κ. Thereafter, for every edge e there are two variables e_κ and $e_{\kappa'}$ by Lemma 9. The NIC-planar extension adds the term $\neg e_\kappa \vee \neg e_{\kappa'}$. We construct η^* as follows: For every edge e keep the first occurrence e_κ in η and replace the second occurrence $e_{\kappa'}$ by $\neg e_\kappa$ and omit the NIC-extensions. This transformation preserves the equivalence between η^+ and η^*. Now each variable and its negation occur exactly once in η^* and it is a conjunction of terms, which are clauses with 2 to 4 variables or of the form $(a \wedge b) \vee (c \wedge d)$. □

We wish to check the satisfiability of formulas with the properties of Lemma 11. However, they do not satisfy any case for which an efficient algorithm is known. Every variable x or its negation $\neg x$ occur in a clause with a disjunction or η^* is unsatisfiable.

Terms of the form $(a \wedge b) \vee (c \wedge d)$ can be transformed into $(a \vee c) \wedge (a \vee d) \wedge (b \vee c) \wedge (b \vee d)$, which is in 2CNF. However, the variables now occur twice. Two clauses $(x_1 \vee \ldots \vee x_p)$ and $(y_1 \vee \ldots \vee y_q)$ with $y_1 = \neg x_1$ are equivalent to $(x_2 \vee \ldots \vee x_p \vee y_2 \vee \ldots \vee y_q)$, since x_1 and $\neg x_1$ do not occur elsewhere in η^*. Hence, a variable is removed at the expense of a longer clause. However, combining two terms of the form $(a \wedge b) \vee (c \wedge d)$ and $(\neg a \vee x_1 \vee \ldots \vee x_k)$ does not lead to a useful simplification.

As before, there are at most $O(n)$ kites such that the length of η is linear in the size of the given graph G. By Lemma 9 each edge e induces at most two variables e_κ and $e_{\kappa'}$, which are replaced by e and $\neg e$ in η^*.

Theorem 4 *Triangulated NIC-planar graphs can be recognized in cubic time if they are (i) K_5-free or (ii) $X4W$-free.*

Proof: Algorithm $\mathcal{B}_{NIC}$ checks whether an input graph G is a triangulated 1-planar graph and constructs a 1-planar embedding $\mathcal{E}(G)$ and a boolean formula η

so that there is a NIC-planar embedding if and only if the NIC-planar extension η^+ is satisfiable according to Theorem. 2. If G is K_5-free and there is no separating edge, then the computed boolean formula can be transformed into a 2SAT formula, whose satisfiability can be solved in linear time [23]. The case of separating edges is solved by a series of 2SAT problems as in the proof of Lemma 6. Let $\{a, b\}$ be a separating edge with separating crossable edges $f_i = \{x_i, y_i\}$ for $k \geq 2$ and $i = 1, \ldots, k$ with an outer component G_k and inner components $G_1, \ldots, G_{k-1}$. If there is a NIC-planar embedding so that an inner edge f_i with $2 \leq i < k$ can be crossed, then the inner components are removed and are replaced by a 4-cycle $C = (a, x_1, b, y_k)$ with chord $\{a, b\}$. The subformula of the inner components is removed from η. Otherwise, if f_1 or f_k must be crossed, then the inner components are removed and are replaced by a 4^+-wheel graph with vertices a, x_1, b, y_k and z, where z is in the interior of the cycle (a, x_1, b, y_k) and there is a chord $\{a, b\}$. The subformula for the separating edge is replaced by $(e_1 \vee e_3) \wedge (e_1 \vee e_4) \wedge (e_2 \vee e_3) \wedge (e_2 \vee e_4)$, where e_1, e_2, e_3, e_4 are variables corresponding to the edges $\{a, x_1\}, \{b, x_1\}, \{a, y_k\}, \{b, y_k\}$. Since the subformulas for separating edges are evaluated only once, the satisfiability test takes linear time.

In the second case, the boolean formula η is in CNF, since subterms of the form $(a \wedge b) \vee (c \wedge d)$ or $(a \wedge b) \vee c$ are excluded and each variable x and its negation $\neg x$ occur exactly once in η^*. Now a subformula consisting of two clauses $(x \vee \beta)$ and $(\neg x \vee \gamma)$ is equivalent to a single clause $(\beta \vee \gamma)$. This transformation removes one variable and leads to a shorter formula in CNF. Each transformation takes constant time by linking lists. Then η^+ and η^* are satisfiable if and only if the transformation leads to $(x \vee \neg x)$ for some variable x.

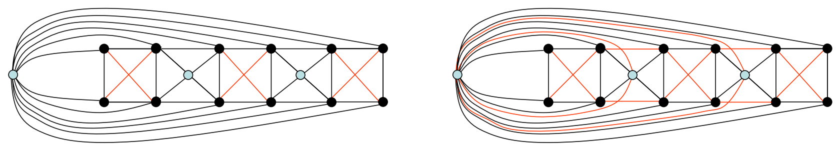
The running time is dominated by the cubic running time of algorithm $\mathcal{B}_{NIC}$. □

Note that 4-connected graphs are K_5-free so that triangulated 4-connected NIC-planar graphs can be recognized in cubic time.

Instances of graphs with complex formulas can be obtained as follows: consider a planar graph H whose faces are triangles or quadrangles. Insert two vertices into each triangle and construct a K_5. Augment each quadrangle to a kite, a 4^+-wheel graph or a double 4^+-wheel graph. Thereby, we obtain a 1-planar embedding $\mathcal{E}(G)$. Now it remains to decide whether there is a NIC-planar embedding $\mathcal{E}'(G)$ such that the edges of H are non-crossed and form separating 3- or 4-cycles.

6.3 Components and Embeddings

We can generalize Theorems 3 and 4 to graphs G, whose 3-connected components are triangulated IC-planar and NIC-planar, respectively. Consider a separation pair $\{u, v\}$ of G and components $H_1, \ldots, H_r$ for some $r > 1$, which each contain the vertices u and v and the edge $e = \{u, v\}$. Then G is IC-planar if and only if each H_i for $i = 1, \ldots, r$ is IC-planar and each of u and v occurs in at most one kite. This is checked as follows: Let algorithm $\mathcal{B}_{IC}$ return the



(a) (b)

Figure 17: ((a) IC planar graphs and (b) (optimal) NIC-planar graphs with exponentially many embeddings. Each 5-wheel graph with an open inner vertex can be flipped, which implies a change of crossed and non-crossed edges in the adjacent kites. Such flips do not change the picture.

boolean formula η_i on H_i. If η_i has a variable x_κ for $x \in \{u, v\}$ and some kite κ, then set $x_\kappa = \mathsf{false}$. If thereafter the IC-extension η_i^+ is not satisfiable, then vertex x is needed in H_i. Graph G is not IC-planar if there are at least two such H_i. Accordingly, if v is an articulation vertex with components $J_1, \ldots, J_s$ for some $s > 1$, then check each component $J_i + v$ and check that v is in a kite of at most one component. Clearly, a graph is IC-planar if so are its disconnected components.

Similarly, G is NIC-planar if and only if each 3-connected component H_i is NIC-planar and the edge $\{u, v\}$ between the vertices of a separation pair u and v occurs in at most one kite, which is checked as before.

Note that the decomposition of a graph at its separation pairs corresponds to the introduction of holes in maps [8]. Alternatively, one may apply the algorithm from [9] and first augment a kite-augmented NIC-planar (IC-planar) graph to a triangulated graph. We can summarize:

Theorem 5 *There is a cubic-time algorithm to test whether a graph is triangulated IC-planar (K_5-free or $X4W$-free triangulated NIC-planar) if so is each 3-connected component.*

The algorithms $\mathcal{B}$, $\mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$ color the edges of a triangulated 1-planar graph such that black, brown and grey edges are always embedded non-crossed, red, blue and orange edges are crossed, and the status of cyan edges is open. Nevertheless, the algorithms cannot determine an embedding. There are triangulated IC-planar and NIC-planar graphs with exponentially many embeddings, as shown in Fig. 17. These graphs are maximal in their class. If the algorithms $\mathcal{B}_{IC}$ and $\mathcal{B}_{NIC}$ applied to these graphs, they find separating 3-cycles and many small graphs with a K_5, which each have two embeddings.

7 Conclusion

We have shown that triangulated IC-planar and K_5-free or $X4W$-free NIC-planar graphs can be recognized in cubic-time. On the other hand, the general recognition problem for IC-planar and NIC-planar graphs is NP-hard. We claim that it remains NP-hard if the graphs are 3-connected and are given with a rotation system that describes the cyclic order of the edges incident to each vertex, as in the case of 1-planar [4] and IC-planar graphs [11]. We also claim that triangulated NIC-planar graphs can be recognized in polynomial time. Optimal 1-planar [10] and optimal NIC-planar graphs [5] can be recognized in linear time, whereas a related result is open for optimal IC-planar graphs.

8 Acknowledgement

I wish to thank Christian Bachmaier and the anonymous reviewers for many valuable suggestions.

References

[1] M. J. Alam, F. J. Brandenburg, and S. G. Kobourov. Straight-line drawings of 3-connected 1-planar graphs. In S. Wismath and A. Wolff, editors, *GD 2013*, volume 8242 of *LNCS*, pages 83–94. Springer, 2013. doi:10.1007/978-3-319-03841-4_8.

[2] M. Albertson. Chromatic number, independence ratio, and crossing number. *Ars Math. Contemp.*, 1(1):1–6, 2008.

[3] C. Auer, C. Bachmaier, F. J. Brandenburg, A. Gleißner, K. Hanauer, D. Neuwirth, and J. Reislhuber. Outer 1-planar graphs. *Algorithmica*, 74(4):1293–1320, 2016. doi:10.1007/s00453-015-0002-1.

[4] C. Auer, F. J. Brandenburg, A. Gleißner, and J. Reislhuber. 1-planarity of graphs with a rotation system. *J. Graph Algorithms Appl.*, 19(1):67–86, 2015. doi:10.7155/jgaa.00347.

[5] C. Bachmaier, F. J. Brandenburg, K. Hanauer, D. Neuwirth, and J. Reislhuber. NIC-planar graphs. *Discrete Appl. Math.*, 232:23–40, 2017. doi:10.1016/j.dam.2017.08.015.

[6] M. J. Bannister, S. Cabello, and D. Eppstein. Parameterized complexity of 1-planarity. In *WADS 2013*, volume 8037 of *LNCS*, pages 97–108. Springer, 2013. doi:10.1007/978-3-642-40104-6_9.

[7] R. Bodendiek, H. Schumacher, and K. Wagner. Über 1-optimale Graphen. *Mathematische Nachrichten*, 117:323–339, 1984. doi:10.1002/mana.3211170125.

[8] F. J. Brandenburg. On 4-map graphs and 1-planar graphs and their recognition problem. *CoRR*, abs/1509.03447, 2015. http://arxiv.org/abs/1509.03447.

[9] F. J. Brandenburg. Characterizing and recognizing 4-map graphs. *Algorithmica, to appear*, 2018.

[10] F. J. Brandenburg. Recognizing optimal 1-planar graphs in linear time. *Algorithmica*, 80(1):1–28, 2018. URL: doi:10.1007/s00453-016-0226-8.

[11] F. J. Brandenburg, W. Didimo, W. S. Evans, P. Kindermann, G. Liotta, and F. Montecchianti. Recognizing and drawing IC-planar graphs. *Theor. Comput. Sci.*, 636:1–16, 2016. doi:10.1016/j.tcs.2016.04.026.

[12] F. J. Brandenburg, D. Eppstein, A. Gleißner, M. T. Goodrich, K. Hanauer, and J. Reislhuber. On the density of maximal 1-planar graphs. In W. Didimo and M. Patrignani, editors, *GD 2012*, volume 7704 of *LNCS*, pages 327–338. Springer, 2013. doi:10.1007/978-3-642-36763-2_29.

[13] S. Cabello and B. Mohar. Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM J. Comput.*, 42(5):1803–1829, 2013. doi:10.1137/120872310.

[14] Z. Chen, M. Grigni, and C. H. Papadimitriou. Planar map graphs. In *Proc. 30th Annual ACM Symposium on the Theory of Computing*, pages 514–523, 1998. doi:10.1145/276698.276865.

[15] Z. Chen, M. Grigni, and C. H. Papadimitriou. Map graphs. *J. ACM*, 49(2):127–138, 2002. doi:10.1145/506147.506148.

[16] Z. Chen, M. Grigni, and C. H. Papadimitriou. Recognizing hole-free 4-map graphs in cubic time. *Algorithmica*, 45(2):227–262, 2006. doi:10.1007/s00453-005-1184-8.

[17] N. Chiba and T. Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14(1):210–223, 1985. doi:10.1137/0214017.

[18] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[19] W. Didimo. Density of straight-line 1-planar graph drawings. *Inform. Process. Lett.*, 113(7):236–240, 2013. doi:10.1016/j.ipl.2013.01.013.

[20] R. Diestel. *Graph Theory*. Springer, 2000.

[21] P. Eades and G. Liotta. Right angle crossing graphs and 1-planarity. *Discrete Applied Mathematics*, 161(7-8):961–969, 2013. doi:10.1016/j.dam.2012.11.019.

[22] R. B. Eggleton. Rectilinear drawings of graphs. *Utilitas Math.*, 29:149–172, 1986.

[23] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[24] A. Grigoriev and H. L. Bodlaender. Algorithms for graphs embeddable with few crossings per edge. *Algorithmica*, 49(1):1–11, 2007. doi:10.1007/s00453-007-0010-x.

[25] S. Hong, P. Eades, N. Katoh, G. Liotta, P. Schweitzer, and Y. Suzuki. A linear-time algorithm for testing outer-1-planarity. *Algorithmica*, 72(4):1033–1054, 2015. doi:10.1007/s00453-014-9890-8.

[26] M. Kaufmann and D. Wagner. *Drawing Graphs*, volume 2025 of *LNCS*. Springer, 2001.

[27] S. G. Kobourov, G. Liotta, and F. Montecchiani. An annotated bibliography on 1-planarity. *Computer Science Review*, 2017. doi:10.1016/j.cosrev.2017.06.002.

[28] V. P. Korzhik and B. Mohar. Minimal obstructions for 1-immersion and hardness of 1-planarity testing. *J. Graph Theor.*, 72:30–71, 2013. doi:10.1002/jgt.21630.

[29] D. Král and L. Stacho. Coloring plane graphs with independent crossings. *Journal of Graph Theory*, 64(3):184–205, 2010. doi:10.1002/jgt.20448.

[30] J. Kyncl. Enumeration of simple complete topological graphs. *Eur. J. Comb.*, 30(7):1676–1685, 2009. doi:10.1016/j.ejc.2009.03.005.

[31] G. Ringel. Ein Sechsfarbenproblem auf der Kugel. *Abh. aus dem Math. Seminar der Univ. Hamburg*, 29:107–117, 1965. doi:10.1007/bf02996313.

[32] H. Schumacher. Zur Struktur 1-planarer Graphen. *Mathematische Nachrichten*, 125:291–300, 1986.

[33] R. Tamassia, editor. *Handbook of Graph Drawing and Visualization.* CRC Press, 2013.

[34] C. Thomassen. Rectilinear drawings of graphs. *J. Graph Theor.*, 12(3):335–341, 1988. doi:10.1002/jgt.3190120306.

[35] M. Thorup. Map graphs in polynomial time. In *Proc. 39th FOCS*, pages 396–405. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743490.

[36] X. Zhang. Drawing complete multipartite graphs on the plane with restrictions on crossings. *Acta Math. Sinica, English Series*, 30(12):2045–2053, 2014.

[37] X. Zhang and G. Liu. The structure of plane graphs with independent crossings and its application to coloring problems. *Central Europ. J. Math*, 11(2):308–321, 2013.