Charles University in Prague
Faculty of Mathematics and Physics

Habilitation Thesis

Drawings of Graphs

RNDr. Vít Jelínek, Ph.D.

Computer Science Institute
Prague, April 1, 2016

Contents

Preface

This habilitation thesis is a commented collection of nine research papers that I authored in the years 2007–2015 with various groups of collaborators. The papers are listed below in chronological order, together with a citation that points to the version of each paper that was included in the thesis. Apart from the papers themselves, the thesis includes three introductory chapters, which provide a brief overview of the area of graph drawings, and outline the main results obtained in the papers.

Papers published in conference proceedings need to conform to strict page limits, and therefore they often omit some parts of the results or proofs. This is the case of papers number 2, 7 and 9 below. In those cases, the citation also includes a reference to a longer preprint version of the same paper, where the interested reader may find the parts of the text that were omitted from the proceedings (and hence also from this thesis).

1. J. Černý, Z. Dvořák, V. Jelínek, and J. Kára: Noncrossing Hamiltonian paths in geometric graphs. *Discrete Applied Mathematics*, 155(9):1096–1105, 2007. See Appendix A.

2. V. Jelínek, E. Jelínková, J. Kratochvíl, and B. Lidický: Clustered planarity: Embedded clustered graphs with two-component clusters. In Proceedings of 16th International Symposium on Graph Drawing, *Lecture Notes in Computer Science* 5417, 121–132, 2009. See Appendix B. Full version: `http://iuuk.mff.cuni.cz/~jelinek/pubs/TwoComponent.pdf`.

3. V. Jelínek, O. Suchý, M. Tesař, and T. Vyskočil: Clustered planarity: Clusters with few outgoing edges. In Proceedings of 16th International Symposium on Graph Drawing, *Lecture Notes in Computer Science* 5417, 102–113, 2009. See Appendix C.

4. V. Jelínek, J. Kynčl, R. Stolař, and T. Valla: Monochromatic triangles in two-colored plane. *Combinatorica*, 29(6):699–718, 2009. See Appendix D.

5. V. Jelínek, E. Jelínková, J. Kratochvíl, B. Lidický, M. Tesař, and T. Vyskočil: The planar slope number of planar partial 3-trees of bounded degree. *Graphs and Combinatorics*, 29(4):981–1005, 2013. See Appendix E.

6. V. Jelínek, J. Kratochvíl, and I. Rutter: A Kuratowski-type theorem for planarity of partially embedded graphs. *Computational Geometry*, 46(4):466–492, 2013. See Appendix F.

7. G. Da Lozzo, V. Jelínek, J. Kratochvíl, and I. Rutter: Planar embeddings with small and uniform faces. In Proceedings of 25th International Symposium on Algorithms and Computation, *Lecture Notes in Computer Science* 8889, 633–645, 2014. See Appendix G. Full version: `http://arxiv.org/abs/1409.4299`.

8. P. Angelini, G. Di Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter: Testing planarity of partially embedded graphs. *ACM Transactions on Algorithms*, 11(4), Article No. 32, 42 pages, 2015. See Appendix H.

9. M. Balko, V. Jelínek, P. Valtr, and B. Walczak: On the Beer index of convexity and its variants. In Proceedings of the 31st International Symposium on Computational Geometry, *Leibniz International Proceedings in Informatics* 34, 406–420, 2015. See Appendix I. Full version: `http://arxiv.org/abs/1412.1769`.

Chapter 1

The Topic of this Thesis

In discrete mathematics, a *graph* is defined as an ordered pair $G = (V, E)$, where V is an arbitrary set, whose elements are called *vertices*, and E is a set of 2-element subsets of E, called *edges*. This definition, while convenient in its generality, is rather abstract, while human intuition needs a more concrete visualization of a graph. The most common way of visualizing a graph is to draw its vertices as points in a plane and its edges as curves connecting the corresponding vertices.

Formally speaking, a *drawing* of a graph $G = (V, E)$ is a mapping that associates to a every vertex $x \in V$ a distinct point p_x in the plane, and to every edge $e = \{x, y\} \in E$ a continuous curve c_e with endpoints p_x and p_y, and such that for any other vertex $z \in V \setminus \{x, y\}$ the point p_z does not belong to c_e. By a slight abuse of terminology, we will refer to the points p_x as vertices of the drawing, and to the curves c_e as edges of the drawing.

In some circumstances, it is convenient to assume further non-degeneracy properties in a drawing, e.g., that any two distinct edges c_e, c_f share at most finitely many points, that no three edge share an internal point, or that no edge crosses itself.

An important special case of graph drawings are *planar* drawings, in which no two edge-curves cross or touch. Graphs admitting such drawings are known as *planar graphs*. The *face* of a planar drawing is a maximal subset of the plane disjoint from all the vertices and edges of the drawing.

Graph drawings are not just a helpful tool in reasoning about graphs, but also also an object of mathematical study in their own right. Various aspects of graph drawings are studied from the point of view of computational complexity, extremal combinatorics, topology, Ramsey theory or discrete geometry.

This thesis is a collection of nine research papers related to the study of graph drawings. The papers included in this thesis can be classified into two groups. The first group, which will be introduced in more detail in Chapter 2, deals exclusively with planar drawings. More specifically, it includes five papers on the topic of algorithmic and structural aspects of planar drawings satisfying certain additional constraints.

The second group, to be introduced in Chapter 3, is focused on the so-called *geometric drawings*, which are drawings (not necessarily planar) where every edge is represented by a straight-line segment. The second group covers a broader range of topics than the first one. The four papers included in this group deal not only with algorithmic aspects of graph drawings, but also with problems related to extremal graph theory, Ramsey theory, and geometry.

Chapter 2

Constrained Planar Drawings

The research of planar graphs has a long history. In 1930, Kuratowski [43] proved that a graph is planar if and only it does not contain a subdivision of K_5 or $K_{3,3}$ as subgraphs. An analogous characterization in terms of forbidden minors was later given by Wagner [58]. Both these characterizations imply that the problem of determining whether a graph is planar can be solved in polynomial time, by using the general results of Robertson and Seymour [49]. However, an algorithm based on this approach would be impractical and non-constructive.

The first linear-time algorithm for planarity testing was given by Hopcroft and Tarjan [33]. Later, other linear-time planarity tests were published by Booth and Lueker [12], Boyer and Myrvold [13], or de Fraysseix et al. [18], among others. These algorithms are constructive, in the sense that they compute, for a given graph G, an explicit planar drawing if it exists. Thus, from algorithmic point of view, planarity testing is a solved problem.

In many practical situations, our task is to find for a given graph G a planar drawing satisfying certain additional restrictions. For instance, we may wish for some groups of vertices to be drawn 'close to each other' to reflect some underlying structure of the vertex set. Or we may insist that the drawing satisfies some aesthetic criteria, e.g., that its faces have approximately the same length.

In this chapter, we will consider several such constraints and we will look at the complexity issues and structural obstacles related to the existence of the corresponding constrained planar drawings.

Since a given planar graph $G = (V, E)$ has infinitely many possible planar drawings, the algorithms dealing with planarity-related problems cannot work with drawings directly. Instead, they must represent the drawings by an appropriate combinatorial description. Often, the combinatorial description of a drawing consists of the so-called *rotation scheme*, which assigns to every vertex $x \in V$ a cyclic order of the edges incident to x, specifying the clockwise circular order in which the edges will appear around the point representing the vertex x.

For a connected planar graph G, a rotation scheme determines the planar drawing of G uniquely up to the choice of the outer face and a continuous deformation of the plane. To avoid the ambiguity in the choice of the outer face, we may select two consecutive edges in a rotation scheme of a vertex, and specify that the common face adjacent to these two edges is the outer face.

Alternatively, we may work with drawings on the surface of the sphere rather than in the plane, avoiding the issue of outer face entirely.

For a disconnected graph G, the rotation scheme may not be sufficient to describe a drawing, since it does not specify the mutual position of the connected components of the graph. A possible approach is then to specify, apart from the rotation scheme, the incidence relation between edges and faces of the drawing.

In the high-level overview presented in this chapter, we always omit the technical details of the particular combinatorial representation of the drawing used in a given algorithm. We also make the convention that the term 'drawing' may refer either to a drawing itself or to its combinatorial representation, depending on context.

Although the problems tackled in the area of constrained planarity are quite diverse, there are certain methods and tools that appear to be useful in many different contexts. One such useful tool is the so-called *SPQR-tree*, introduced by Di Battista and Tamassia [20]. The SPQR-tree of a biconnected planar graph G is a tree-like data structure, which can be computed in linear time [32], and which represents in a compact way all possible planar drawings of G. The definition of the SPQR-tree is somewhat technical, and can be found in Appendix H, Section 2.3.

Another general approach, which has received a lot of attention recently, is based on the Hanani–Tutte theorem [55]. The theorem states that if a graph G has a drawing in which any pair of vertex-disjoint edges is represented by curves that cross an even number of times, then G also has a planar drawing. Several authors [28, 29, 47, 51, 52] have recently discovered generalizations of this result applicable to various constrained types of planar drawings. This approach seems to offer a promising area of further research.

2.1 Partially Embedded Planarity

One of the most natural questions we may ask in the context of constrained planarity testing is the following: suppose that we are given a graph G, and suppose that a planar drawing has been prescribed for a subgraph H of G; is it possible to complete the drawing of H into a planar drawing of G?

To state the question more precisely, we introduce the following concepts: a *partially embedded graph* (or PEG) is a triple $(G, H, \mathcal{H})$, where G is a graph, H is a subgraph of G, and $\mathcal{H}$ is a planar drawing of H. A PEG is said to be *planar* if there is a planar drawing of G that extends the given drawing $\mathcal{H}$ of H.

The problem of deciding whether a given PEG is planar is known as *partially embedded planarity* (or PEP). The first and most obvious question that has been asked in connection with PEP was whether the problem admits an efficient algorithmic solution.

It is clear that for a PEG $(G, H, \mathcal{H})$ to be planar, it is necessary for G itself to be a planar graph. However, this condition is far from sufficient, as can be seen on simple examples of non-planar PEGs with planar underlying graphs (see Figure 2.1).

Moreover, while there are several polynomial algorithms for testing planarity of graphs, there is no known way to adapt any of them into an efficient approach to test partially embedded planarity. For an illustration of the difference between planarity testing and PEP-testing, notice that a PEG $(G, H, \mathcal{H})$ may be

Figure 2.1: Two examples of a non-planar Peg whose underlying graph is planar. The solid lines represent the edges of the embedded part, while the dashed curves indicate the non-embedded edges.

non-planar even when every connected component of G induces a planar Peg, as demonstrated by the example in the right part of Figure 2.1.

In a joint paper with Angelini, Di Battista, Frati, Kratochvíl, Patrignani and Rutter, titled '**Testing planarity of partially embedded graphs**' [4, see also Appendix H], we have proved that Pep can be solved in polynomial time.

Our construction of a polynomial-time test for Pep proceeds in several steps. First, we argue that testing planarity of a general Peg $(G, H, \mathcal{H})$ can be efficiently reduced to the situation when the graph G is 2-connected. The general idea here is to test planarity of each biconnected component of $(G, H, \mathcal{H})$ separately, and then verify certain technical but efficiently testable properties of $(G, H, \mathcal{H})$ that characterize whether the drawings of individual biconnected components can be glued together into a drawing of the whole Peg.

To test planarity of a Peg $(G, H, \mathcal{H})$ with G biconnected, we construct the SPQR tree of G, and use it to characterize planar Pegs. Vaguely speaking, our characterization says that a Peg $(G, H, \mathcal{H})$ with biconnected underlying graph G is planar if and only if each skeleton in the SPQR tree of G admits a planar drawing that is 'compatible' with the partial drawing $\mathcal{H}$. Here again, the precise meaning of being 'compatible' is too technical to be described precisely, but the important fact is that the existence of a compatible drawing may be tested in polynomial time, yielding a polynomial algorithm for Pep. The algorithm is constructive, i.e., it constructs a planar drawing of G extending $\mathcal{H}$, if such a drawing exists.

In our paper, we further refine our approach to obtain a Pep-testing algorithm that runs in linear time. The algorithm itself, as well as the analysis of its time complexity, are rather nontrivial.

A different aspect of partially embedded planarity is studied in the paper '**A Kuratowski-type theorem for planarity of partially embedded graphs**' [34, see Appendix F], which is a joint work with Kratochvíl and Rutter. Our goal in this paper is to characterize planar Pegs by minimal forbidden substructures, in analogy to the classical theorems of Kuratowski and Wagner.

To obtain such a characterization for partially embedded planarity, we first introduce a suitable partial order relation on the set of Pegs that is analogous to the graph minor relation. In particular, this Peg-minor relation is be chosen in such a way that every Peg-minor of a planar Peg is again planar.

In our paper, we actually consider two possible versions of the Peg-minor relation. The first one is a straightforward adaptation of the graph minor relation. As our main result, we are able to describe all the minimal non-planar

PEGs with respect to this relation. Unfortunately, it turns out that there are infinitely many such minimal non-planar PEGs. To reduce the set of minimal non-planar PEGs even further, we then introduce a stronger PEG-minor relation, which is defined using a larger set of planarity-preserving operations, and which only admits a finite number of minimal non-planar PEGs.

We remark that our characterization of partially embedded planarity via minimal forbidden obstructions has been subsequently used by Schaefer [51] to prove an analogue of the Hanani–Tutte theorem for this version of planarity.

We also note that Patrignani [48] has considered an analogue of partially embedded planarity in the setting of geometric planar graphs, i.e., of planar drawings whose every edge is drawn as a straight-line segment. In contrast to our results, he showed that for a given PEG $(G, H, \mathcal{H})$ where $\mathcal{H}$ is a geometric planar graph, it is NP-hard to decide whether $\mathcal{H}$ can be extended into a geometric planar drawing of G.

2.2 Clustered Planarity

Let $G = (V, E)$ be graph. A *cluster hierarchy* is a set $\mathcal{C}$ of subsets of V, called *clusters*, with the property that any two clusters in $\mathcal{C}$ are either disjoint or one is a subset of the other. A *clustered graph* is a graph G together with its cluster hierarchy $\mathcal{C}$.

Given a clustered graph $(G, \mathcal{C})$, our goal is to find a planar drawing of G in which the vertices of each cluster are 'grouped together' to display the cluster hierarchy. This is formalized by the concept of *clustered-planar drawing* (or c-planar drawing, for short). A c-planar drawing of $(G, \mathcal{C})$ is a planar drawing of G together with a function γ which associates with every cluster $X \in \mathcal{C}$ a simply connected compact region γ_X satisfying these properties (see Figure 2.2):

- If X and Y are disjoint clusters, then γ_X and γ_Y are disjoint as well, and if $X \subseteq Y$ then $\gamma_X \subseteq \gamma_Y$.
- For every cluster $X \in \mathcal{C}$, all the vertices of X are drawn in the interior of γ_X, while all the vertices of $V \setminus X$ are drawn outside of γ_X.
- For every edge $e = uv$ and every cluster X, the drawing of uv intersects the boundary of γ_X at most once, and in such case the intersection is a proper crossing. In particular, this means that if both u and v belong to X, then the edge e is drawn entirely inside γ_X, while if neither u nor v belongs to X, then the drawing of e is disjoint from γ_X.

The key problem related to c-planarity is to decide whether a given clustered graph has a c-planar drawing. There are in fact two versions of the corresponding decision problem. In the first version, which is the so-called *non-embedded c-planarity*, we are given a clustered graph $(G, \mathcal{C})$ and the task is to decide whether $(G, \mathcal{C})$ has a c-planar drawing. In the other version, known as *embedded c-planarity*, we are given a clustered graph $(G, \mathcal{C})$ and a planar drawing of G, and the task is to decide whether the given drawing of G can be extended into a c-planar drawing by a suitable choice of cluster regions.

It is not hard to show that both versions of c-planarity lie in the complexity class NP. It is, however, a major open problem to determine whether these problems can be solved in polynomial time.

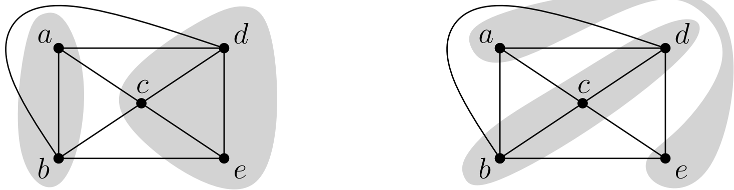


Figure 2.2: Left: a c-planar drawing of a clustered graph with cluster hierarchy $\mathcal{C} = \{\{a, b\}, \{c, d, e\}\}$. The shaded regions represent the two clusters. Right: the same graph with the cluster hierarchy $\mathcal{C} = \{\{b, c, d\}, \{a, e\}\}$ is no longer c-planar. Note that the right part of the figure is not a valid c-planar drawing, since the edge $\{b, d\}$ crosses twice the boundary of the region representing the cluster $\{a, e\}$.

Various authors have found polynomial algorithms solving special cases of c-planarity, in both embedded and non-embedded setting. For instance, non-embedded c-planarity can be solved in polynomial time for instances having only two clusters (Biedl [9]), instances where each cluster induces a connected subgraph (Lengauer [44], and independently Feng et al. [27]), or instances whose underlying graph is a cycle and each cluster has size at most three (Jelínková et al. [39]). Despite these partial results, the complexity of c-planarity testing remains open even for seemingly simple special cases, e.g., when the underlying graph is a cycle, or when the instance only has three clusters.

For embedded c-planarity testing, the situation is similar. Again, there are polynomial algorithms known for special instances. For example, Di Battista et al. [19] have found an algorithm for embedded c-planarity for instances whose underlying graph has at most five vertices per face and all clusters are disjoint. Another partial result, due to Chimani et al. [15], gives a polynomial algorithm for instances in which any two clusters are disjoint and each face is incident to at most two vertices from each cluster.

This thesis includes two papers on the topic of c-planarity testing. The first one is titled '**Clustered planarity: Clusters with few outgoing edges**' [38, see also Appendix C], and is a joint work with Suchý, Tesař and Vyskočil. In the paper, we consider the complexity of non-embedded clustered planarity in the situation when each cluster C has a bounded number of *out-going* edges, which are edges connecting a vertex of C with a vertex outside of C. Our main result is a linear time algorithm for instances where each cluster has at most four out-going edges.

Let us sketch the basic idea of our approach. Suppose that $(G, \mathcal{C})$ is a clustered graph with at most four out-going edges per cluster, and suppose that $C \in \mathcal{C}$ is a minimal cluster, i.e., a cluster that has no proper subcluster in $\mathcal{C}$. Let G_C be the subgraph of G induced by the vertices in C.

To construct a c-planar drawing of $(G, \mathcal{C})$, we may first find a planar drawing of G_C in which the vertices incident with the out-going edges of C are drawn on the outer face, and then extend this drawing to a c-planar drawing of G. There can of course be many drawings of G_C; however, when extending a given drawing of G_C into a c-planar drawing of G, the only information about the drawing of G_C we need to take into account is the cyclic order in which the out-going edges of C can be attached around the outer face of G_C.

Thus, for the graph G_C, we first compute the set of all possible cyclic orders of the out-going edges that may arise from planar drawings of G_C. This computation can be done efficiently using standard techniques from the theory of planarity testing. We next remove the cluster C from the cluster hierarchy, and replace G_C with a connected gadget of constant size that imposes the same restrictions on the possible ordering of outgoing edges as G_C. This yields an equivalent instance of c-planarity with fewer clusters (and usually with fewer vertices as well). Iterating this procedure, we eventually reduce the original c-planarity instance to an equivalent planarity instance.

Our simple approach does not generalize to instances with five or more out-going edges per cluster, since there are disconnected graphs G_C whose restrictions on out-going edges cannot be simulated by connected gadgets. However, Bläsius and Rutter [11] have recently shown that c-planarity is tractable even with at most five out-going edges per cluster. Their approach uses non-trivial previous results on the so-called simultaneous PQ-orderings [10]. The complexity status of c-planarity with at most six out-going edges is still open.

The other paper on c-planarity included in this thesis is called '**Clustered Planarity: Embedded Clustered Graphs with Two-Component Clusters**' [35, see Appendix B], and is a joint work with Jelínková, Kratochvíl and Lidický. In this paper, we give a polynomial algorithm for embedded c-planarity of instances in which each cluster induces a subgraph with at most two connected components.

Even prior to this work, it was well understood in the research community that connectivity assumptions may help in dealing with c-planarity. One of the earliest results on c-planarity, due to Feng et al. [27], is a c-planarity testing algorithm (in both embedded and non-embedded settings) for instances in which each cluster induces a connected subgraph. Subsequently, several authors have investigated possibilities for relaxing the connectivity assumption. Thus, Gutwenger et al. [31] have shown that c-planarity is polynomial when all the disconnected clusters belong to a single path in the cluster hierarchy, and also when each disconnected cluster has a connected parent and siblings in the cluster hierarchy, while Goodrich et al. [30] gave an algorithm for instances where each disconnected cluster has a connected parent, and moreover, each component of a disconnected cluster is connected by an edge to a vertex not belonging to the parent cluster.

Our approach to solve embedded c-planarity for instances $(G, \mathcal{C})$ with clusters having at most two components is based on the following idea: we transform $(G, \mathcal{C})$ into an equivalent instance $(G^+, \mathcal{C})$ which has all clusters connected. This is achieved by drawing new edges into G without violating planarity, in such a way that every new edge connects vertices from two distinct components of a disconnected cluster. It can be shown, under some technical assumptions, that such a set of new edges exists if and only if the original clustered graph was c-planar. However, the algorithm to find such new edges is highly nontrivial, even for instances where all clusters are disjoint.

Despite the partial results outlined above, the complexity of c-planarity testing remains one of the main open problems in the area of graph drawing.

2.3 Minimizing Face Size

As we mentioned before, the problem of finding a planar drawing of a given planar graph can be solved in linear time by several well-known approaches [12, 13, 18, 33]. However, a graph may admit many different planar drawings, and in many real-life situations different drawings are not all equally suitable. It is then desirable to optimize over the space of all planar drawings according to a predefined 'quality measure'.

Several such measures are related to the *lengths* of faces, where the length of a face f is defined as the number of edges incident to f. For aesthetic reasons, it is natural to try to avoid drawings having large internal faces, or drawings with faces sizes of varying lengths. A very general formalization of this problem was introduced by Mutzel and Weiskircher [45]: in their setting, we are given a graph G and a function assigning costs to the cycles of the graph, and the goal is to find a drawing of G minimizing the total cost of face cycles. Mutzel and Weiskircher have shown that this problem can be transformed into an integer linear programming problem which can in many practical instances be tackled by existing ILP solvers. Their approach is based on dynamic programming and SPQR-trees.

On the other hand, Woeginger [59] has shown that for any $k \geq 4$, it is NP-hard to find a drawing of a given graph that minimizes the number of faces of length at least k. This means, in particular, that the above-mentioned general problem of Mutzel and Weiskircher is NP-hard even when all the cycles have cost 0 or 1.

In a joint paper with Da Lozzo, Kratochvíl and Rutter, titled '**Planar Embeddings with Small and Uniform Faces**' [17, see Appendix G], we consider the complexity of two closely related decision problems: for a fixed integer $k \geq 3$, the problem k-MinMaxFaces asks to determine whether a given graph has a drawing in which all faces have size at most k, and the problem k-UniformFaces asks whether a given graph has a drawing whose faces all have size exactly k.

In the study of these two decision problems, we restrict ourselves to instances when G is biconnected. This ensures that in any drawing of G, all the face boundaries are simple cycles, avoiding an ambiguity in defining the size of non-biconnected face boundaries.

Observe that for a given k, k-UniformFaces can be easily reduced to k-MinMaxFaces. This is because for a given graph $G = (V, E)$, all the drawings of G have the same average face size, namely $2|E|/(|E| - |V| + 2)$, as can be easily deduced from Euler's formula. It follows that G has a drawing with all faces of size exactly k if and only if its average face size is k and there is a drawing with maximum face size k.

In our paper, we were able to determine the complexity of k-MinMaxFaces completely as a function of k, while for k-UniformFaces we determine the complexity for all k except for the two cases $k = 5$ and $k = 8$, which are left open. The results are summarized in Table 2.1.

While the results we obtained for the complexity of k-MinMaxFaces correspond to the intuitively plausible idea that the problem becomes harder as k grows, the corresponding results for k-UniformFaces may seem surprising at first sight. This is due to the fact that for k even, a graph may only have a drawing with all faces of size k if it is bipartite. Due to this aspect, it is

	k-MinMaxFaces	k-UniformFaces
$k = 3$	Polynomial	Polynomial
$k = 4$	Polynomial	Polynomial
$k = 5$	NP-complete	Unknown
$k = 6$	NP-complete	Polynomial
$k = 7$	NP-complete	NP-complete
$k = 8$	NP-complete	Unknown
$k \geq 9$	NP-complete	NP-complete

Table 2.1: The complexity of k-MinMaxFaces and k-UniformFaces for various values of k.

easier to design polynomial algorithms (and harder to obtain hardness proofs) for k-UniformFaces when k is even then when k is odd.

Apart from the decision problems, our paper also addresses the optimization version of MinMaxFaces, i.e., for a given graph G, we want to find a drawing which minimizes the size of the largest face. We design a polynomial approximation algorithm which has an approximation ratio of 6, i.e., it finds a drawing whose largest face is at most six times larger than the largest face in the optimal solution. The algorithm, as well as our polynomial algorithms for k-MinMaxFaces and k-UniformFaces we have designed, are based on dynamic programming and SPQR-trees.

Chapter 3

Geometric Graphs

In this chapter, we focus on the so-called *geometric graphs*, which are (possibly non-planar) drawings of graphs in which every edge is represented by a straight-line segment.

Each of the four papers presented in this chapter approaches the area of geometric graphs from a different perspective, often only loosely related to graph theory; in fact, two of the papers do not make any explicit use of graph-theoretic concepts. Nevertheless, in this chapter, we take care to point out the underlying graph intuition of our results, to emphasize the versatility and usefulness of the concept of geometric graphs.

3.1 Hamilton Paths in Geometric Graphs

An important area in the study of geometric graphs is concerned with extremal problems, that is, with problems to determine, for a given number of vertices n, the largest possible number $m(n)$ of edges in a geometric graph on n vertices that avoids a given configuration of edges. The function $m(n)$ is called *the extremal function* of the configuration.

For instance, Pach and Törőcsik [46] have shown that for any fixed k, a geometric graph on n vertices with no noncrossing matching of size k has at most $O(n)$ edges. Here, we say that a matching (or more generally, any subgraph) in a geometric graph is *noncrossing* if no two of its edges cross each other.

Another important extremal problem is to determine the maximum number of edges in a geometric graph on n vertices with no k pairwise crossing edges. It is conjectured that for any constant k, such a graph can have at most $O(n)$ edges, but this has so far only been proven for $k = 3$ (Agarwal et al. [2]) and $k = 4$ (Ackerman [1]). For general k, Valtr [56] gave an upper bound of order $O(n \log n)$.

Several authors have also considered the extremal function of noncrossing configurations whose size depends on the number of vertices of the graph. In these situations, the extremal function $m(n)$ tends to be close to $\binom{n}{2}$, and it is therefore more natural to estimate the difference $\binom{n}{2} - m(n)$ rather than $m(n)$ itself. In other words, the problem is to find the smallest possible number of edges in the complement of a geometric graph that avoids the given configuration.

For example, Károlyi et al. [40] have shown that for any geometric graph G,

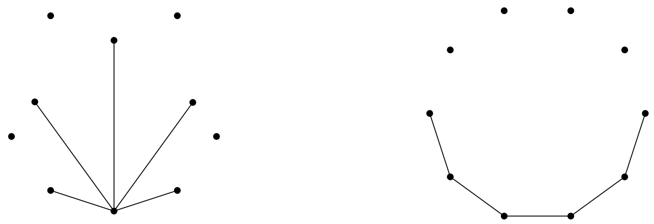


Figure 3.1: Two examples of geometric graphs with n vertices and $\lceil \frac{n}{2} \rceil$ edges whose complement has no noncrossing Hamilton path.

either G or its complement has a noncrossing spanning tree. This implies that every geometric graph whose complement has at most $n-2$ edges has a noncrossing spanning tree, and this is easily seen to be best possible.

The paper '**Non-Crossing Hamiltonian Paths in Geometric Graphs**' [16, see Appendix A] (joint work with Černý, Dvořák and Kára) deals with the following extremal problem, originally asked by Perles: what is the smallest possible number $h(n)$ of edges in the complement of a geometric graph on n vertices that has no noncrossing Hamilton path? A *Hamilton path* in a graph is a path that contains all the graph's vertices.

It is easy to see that every complete geometric graph has a noncrossing Hamilton path; one may obtain such a path, e.g., by traversing the vertices in the left-to-right order. We can push this idea a little further: for $n \geq 3$, every graph on n vertices obtained from the complete graph by the removal of one edge has a noncrossing Hamilton path — we simply rotate the graph in such a way that the two non-adjacent vertices do not appear consecutively in the left-to-right order. This observation shows that $h(n)$ is at least 2 for $n \geq 3$. On the other hand, it is clear that $h(n) \leq n-1$, since a graph whose complement contains $n-1$ edges incident to a single vertex has no Hamilton path. This upper bound can be improved to $h(n) \leq \lceil \frac{n}{2} \rceil$, by a straightforward generalization of the examples depicted on Figure 3.1.

Perles has asked whether the function $h(n)$ is unbounded, and if so, what is its order of growth. In our above-mentioned paper [16], we proved that $h(n)$ is indeed unbounded, as a corollary to the following result.

Theorem 1. *Let $G = (V, E)$ be a complete geometric graph with n vertices in general position, and let X be a subset of its vertices of size at most $\frac{1}{\sqrt{2}}\sqrt{n}$. Then G has a noncrossing Hamilton path in which no two vertices of X appear consecutively.*

An immediate consequence of this result is the estimate $h(n) > \frac{1}{2\sqrt{2}}\sqrt{n}$. To see this, suppose that G is a complete geometric graph with n vertices in general position, and suppose we remove $\frac{1}{2\sqrt{2}}\sqrt{n}$ edges from G. We can then take X to be the set of vertices incident to at least one removed edge. By Theorem 1, G has a noncrossing Hamilton path in which no two vertices of X are consecutive, and in particular, the path avoids all the removed edges.

We have also shown that the bound $\frac{1}{\sqrt{2}}\sqrt{n}$ in Theorem 1 is best possible up to a constant factor, by presenting an example of a complete geometric graph G on n vertices in general position (actually, in strictly convex position) and a subset X of its vertices of size $3\sqrt{n}$, such that each noncrossing Hamilton path in G has two consecutive vertices of X.

The bounds $\frac{1}{2\sqrt{2}}\sqrt{n} < h(n) \leq \lceil \frac{n}{2} \rceil$ still remain the best known estimates on $h(n)$. Subsequently to our work, the extremal functions of several other types of large noncrossing configurations were considered. In particular, Aichholzer et al. [3] have shown that for any n even, any graph on n vertices whose complement has fewer than $\frac{n}{2}$ edges contains a noncrossing perfect matching, and this bound is best possible. Moreover, Aichholzer et al. have shown that for any $k \geq 2$, a geometric graph on n vertices whose complement has fewer than $\lceil \frac{kn}{2} \rceil$ edges contains a noncrossing tree on $n-k+1$ vertices, and this is again best possible.

3.2 Monochromatic Triangles in the Plane

Another area of geometric graph theory is the study of (typically infinite) graphs whose vertices are points in the plane (or in a more general metric space) and edges are defined by a condition depending on the distance of the two vertices. Perhaps the most famous problem in this area is the so-called Hadwiger–Nelson problem, which asks to determine the smallest number of colors needed to color the points in the plane in such a way that no points at distance one from each other are colored by the same color. Assuming the axiom of choice, the problem can be rephrased in the following equivalent form: what is the largest chromatic number of a finite geometric graph whose every edge is a segment of length one?

It is easy to find examples of unit-distance graphs of chromatic number 4, and it is also not too hard to describe a coloring of the plane by 7 colors with every two points at distance 1 receiving distinct colors. However, despite considerable interest in this problem, these simple bounds have not been improved yet.

The Hadwiger–Nelson problem is the simplest special case of the following general question: suppose we are given a (typically finite) set P of points in the plane; what is the smallest number of colors needed to color the plane in such a way that no rotated and translated copy of P is monochromatic? This type of questions, and its obvious generalization to higher dimensions, is studied in the area known as Euclidean Ramsey theory.

We say that a coloring of the plane *contains* the set P if there is a monochromatic translated and rotated copy of P, otherwise we say that the coloring *avoids* P. The Hadwiger–Nelson problem then asks to find the smallest number of colors in a coloring that avoids the configuration P consisting of two points at distance 1 apart.

What can we say about the avoidance of three-point configurations? Let us use the term (a, b, c)-*triangle* to refer to a configuration of three points forming the vertex set of a triangle with edges of length a, b and c. Ramsey properties of triangles were first studied by Erdős et al. [23–25], who focused on colorings by two colors. They have shown that for any triple of edge-lengths a, b, c satisfying the (possibly degenerate) triangle inequality, a two-coloring of the plane contains an (a, b, c)-triangle if and only if it contains an (a, a, a)-triangle or a (b, b, b)-triangle or a (c, c, c)-triangle. This implies, in particular, that a two-coloring contains a given triangle if and only if it contains its mirror image. It also implies that any two-coloring that avoids at least one triangle must avoid at least one equilateral triangle, and any two-coloring that avoids at least one non-equilateral triangle must avoid equilateral triangles of at least two different edge-lengths.

Erdős et al. [23] have also pointed out that the $(1,1,1)$-triangle is avoided by the two-coloring consisting of parallel half-open strips of width $\sqrt{3}/2$ colored by alternating colors; formally, we color a point (x,y) black if $\lfloor \sqrt{3}x/2 \rfloor$ is odd and white otherwise. They then made the following two conjectures.

Conjecture 2. *[25] Each two-coloring of the plane contains each non-equilateral triangle. Equivalently, each two-coloring of the plane avoids at most one equilateral triangle.*

Conjecture 3. *[25] The coloring consisting of alternating strips of width $\sqrt{3}/2$ is the only two-coloring of the plane that avoids the $(1,1,1)$-triangle, up to a possible rotation, translation and modification of the colors of the points on the boundaries of the strips.*

Observe that Conjecture 3 is a strengthening of Conjecture 2, since the colorings described in Conjecture 3 contain all equilateral triangles except for the $(1,1,1)$-triangle.

These conjectures have motivated a joint paper with Kynčl, Stolař and Valla, '**Monochromatic Triangles in Two-Colored Plane**' [37, see Appendix D]. In this paper, our aim was to verify the above conjectures of Erdős et al. for special types of colorings. As our first result, we have shown that any two-coloring of the plane in which one of the colors forms a topologically closed subset of the plane (and hence the other color is open) contains all triangles, verifying Conjecture 2 for such colorings.

Next, we have considered the so-called *polygonal* colorings, which are colorings of the plane by black and white colors in which the common boundary of the two colors consists of a union of straight-line segments. In this class of colorings, we have found counterexamples to Conjecture 3. More precisely, we defined a family of so-called *zebra-like* colorings, which generalize the colorings described in Conjecture 3 by considering strips which are not bounded by two parallel lines, but rather by a pair of periodic piecewise linear curves satisfying certain constraints.

We also proved that among polygonal colorings, the zebra-like colorings are the only counterexamples to Conjecture 3. In particular, we verified Conjecture 2 for polygonal colorings.

Despite some partial results verifying Conjecture 2 for specific types of triangles (see e.g. [25, 53]), the conjecture remains open to this day.

3.3 Drawings with Few Slopes

A classical result, proven independently by Wagner [57] and Fáry [26], states that every planar graph has a planar geometric drawing. For aesthetic purposes, it is natural to look for a geometric drawing whose edges form as few distinct slopes as possible.

Clearly, if a graph contains a vertex of degree Δ, then its edges must form at least $\lceil \Delta/2 \rceil$ distinct slopes in any geometric drawing. This lower bound is in general not tight, as shown, e.g., by the example of the graph K_4, whose every planar geometric drawing requires six distinct slopes.

Dujmović et al. [21, 22] have asked whether there is a function f such that any planar graph of maximum degree Δ has a planar geometric drawing whose edges determine at most $f(\Delta)$ slopes. They proved that every tree has such a drawing with at most $\lceil \Delta/2 \rceil$ slopes, and that every cubic 3-connected planar graph has such a drawing with at most 6 slopes.

In the paper '**The Planar Slope Number of Planar Partial 3-Trees of Bounded Degree**' [36, see Appendix E] (joint work with Jelínková, Kratochvíl, Lidický, Tesař, and Vyskočil), we obtain analogous results for another class of graphs, namely planar partial 3-trees, i.e., the planar graphs of tree-width at most 3. Specifically, we proved the next theorem.

Theorem 4. *Every planar partial 3-tree of maximum degree Δ has a geometric planar drawing whose edges determine $O(\Delta^5)$ slopes.*

Subsequently, Keszegh et al. [41] have answered the open problem of Dujmović et al. in full generality, by showing that every planar graph of maximum degree Δ has a planar geometric drawing with $2^{O(\Delta)}$ slopes. It is an open problem whether the exponential dependency on Δ can be improved. The best known lower bound is also due to Keszegh et al. [41], who showed that for any Δ, there is a planar graph G_Δ that requires at least $3\Delta - 6$ slopes in any planar geometric drawing. The gap between the upper bound and the lower bound is thus quite large.

For more restricted classes of graphs, tight results are available. For instance, Knauer et al. [42] have shown that every outerplanar graph of maximum degree $\Delta \geq 4$ has an outerplanar straight-line drawing with at most $\Delta - 1$ slopes, and this is best possible, in the sense that for every $\Delta \geq 4$ there is an outerplanar graph of maximum degree Δ whose every outerplanar straight line drawing requires $\Delta - 1$ slopes.

3.4 Visibility and Convexity

Suppose that A is a set in a Euclidean space $\mathbb{R}^d$, and let $\lambda(A)$ denote the (Lebesgue) measure of A. We define the *visibility graph* G_A of the set A to be the (typically infinite) geometric graph whose vertices are the points of A, and two points $x, y \in A$ are connected by an edge if and only if the segment xy is contained in A. Notice that G_A is a clique if and only if the set A is convex.

We may define certain properties of the graph G_A which indicate that the set A is in some sense 'close' to being convex. Suppose, for the rest of this chapter, that the set $A \subseteq \mathbb{R}^d$ is measurable, with finite positive measure, and suppose that the set $\mathrm{Seg}(A) = \{(x, y) \in A \times A;\ xy \text{ is an edge of } G_A\}$ is measurable as well. Let us consider the following two quantities:

- The *convexity ratio* of A, denoted by $\mathrm{c}(A)$, is defined by

$$\mathrm{c}(A) = \sup_{\substack{K \subseteq A \\ K \text{ convex}}} \frac{\lambda(K)}{\lambda(A)}.$$

- The *Beer index* of A, denoted by $\mathrm{b}(A)$, is defined by

$$\mathrm{b}(A) = \frac{\lambda(\mathrm{Seg}(A))}{\lambda(A)^2}.$$

Note that both $\mathrm{b}(A)$ and $\mathrm{c}(A)$ take values from the interval $[0,1]$, and if A is convex, then $\mathrm{b}(A) = \mathrm{c}(A) = 1$. Thus, $\mathrm{b}(A)$ and $\mathrm{c}(A)$ can both be seen as indicators of how close a set is to being convex.

In graph theoretic terms, $\mathrm{b}(A)$ can be interpreted as the edge-density of the graph G_A, while $\mathrm{c}(A)$ is the relative size of the largest clique in A. We may also view $\mathrm{b}(A)$ as the probability that for two points x, y chosen uniformly independently in A, the segment xy is contained in A.

The Beer index was introduced and studied by Beer [6–8], who called it 'the index of convexity'. Later, it was rediscovered by Stern [54], who described a Monte Carlo algorithm to approximate the Beer index of a polygon. Rote [50] then gave an exact algorithm to compute the Beer index of a polygon.

What can we say about the relationship between $\mathrm{b}(A)$ and $\mathrm{c}(A)$? On one hand, it is easy to see that the Beer index of a set A is at least $\mathrm{c}(A)^2$, and this cannot be improved, as demonstrated, for instance, by a set A constructed as a disjoint union of a compact convex set of measure c and a set of measure $1-c$ with Beer index 0.

A more interesting question is to determine whether there is a lower bound on $\mathrm{c}(A)$ in terms of $\mathrm{b}(A)$, or in graph theoretic terms, whether a given constant edge-density of the visibility graph of A implies the graph has a clique forming a constant fraction of A.

Without any further assumptions on A, we cannot get any nontrivial bound of this form, as shown by the set $A = [0,1]^2 \setminus \mathbb{Q}^2$, obtained from a unit square by removing all the points that have both coordinates rational. One may easily check that $\mathrm{b}(A) = 1$ while $\mathrm{c}(A) = 0$.

The situation changes, however, when we impose additional restrictions on A. In particular, Cabello et al. [14] have shown that there is a constant $\alpha > 0$ such that every polygon P satisfies $\mathrm{b}(P) \leq \alpha \cdot \mathrm{c}(P)\,(1 - \log \mathrm{c}(P))$. They conjectured that this can be further strengthened to a linear bound $\mathrm{b}(P) \leq \alpha \cdot \mathrm{c}(P)$, and verified the conjecture for star-shaped polygons, i.e., for polygons containing a point which is adjacent to all the other points in the visibility graph.

In the paper '**On the Beer index of convexity and its variants**' [5, see Appendix I], which is a joint work with Balko, Valtr and Walczak, we confirm and generalize this conjecture. Specifically, we obtain the following result.

Theorem 5. *There is a constant $\alpha > 0$ such that every simply connected set $A \subseteq \mathbb{R}^2$ satisfies $\mathrm{b}(A) \leq \alpha \cdot \mathrm{c}(A)$.*

Note that up to the value of α, the bound in the previous theorem is optimal, as shown by the example of a set A consisting of n internally disjoint triangles of equal area, sharing a common vertex x, and positioned in such a way that x is on the boundary of the convex hull of A. We then see that $\mathrm{b}(A) = \mathrm{c}(A) = 1/n$.

We also consider analogues of these results in higher dimensions, i.e., in the situation when A is a subset of $\mathbb{R}^d$. To state these generalizations, it is convenient to introduce higher-order generalizations of the Beer index. For a set $A \subseteq \mathbb{R}^d$, and $k \leq d$ an integer, we let $\mathrm{b}_k(A)$ denote the probability that for a $(k+1)$-tuple of points $(x_0, x_1, \ldots, x_k)$ chosen uniformly independently at random in A, the convex hull of $x_0, \ldots, x_k$ is a subset of A. Note that with this notation, $\mathrm{b}_1(A)$ is the Beer index $\mathrm{b}(A)$ of A. Note also that we have the inequalities $\mathrm{b}_1(A) \geq \mathrm{b}_2(A) \geq \cdots \geq \mathrm{b}_d(A)$, provided all the $\mathrm{b}_k(A)$ are well defined.

We then obtain the following lower bound for $\mathrm{c}(A)$ in terms of $\mathrm{b}_d(A)$.

Theorem 6. *For every $d \geq 2$ there is a constant $\beta = \beta(d) > 0$ such that every set $A \subseteq \mathbb{R}^d$ whose $\mathrm{b}_d(A)$ is well defined satisfies $\mathrm{b}_d(A) \leq \beta \cdot \mathrm{c}(A)$.*

Compared to Theorem 5, in Theorem 6 we do not require any special properties of A, beyond the assumption that $\mathrm{b}_d(A)$ is well defined. On the other hand, for $d = 2$, Theorem 6 only gives a lower bound for $\mathrm{c}(A)$ in terms of $\mathrm{b}_2(A)$, while Theorem 5 gives a bound in terms of $\mathrm{b}(A) = \mathrm{b}_1(A)$. Thus, the two theorems are incomparable in terms of strength.

We do not know whether the linear bound of Theorem 6 is best possible, even for $d = 2$. However, we have a construction showing that the bound is tight up to a logarithmic factor.

Theorem 7. *For every $d \geq 2$ there is a constant $\gamma = \gamma(d) > 0$ such that for each $\varepsilon \in (0, 1]$ there is a set $A \subseteq \mathbb{R}^d$ with $\mathrm{c}(A) \leq \varepsilon$ satisfying $\mathrm{b}_d(A) \geq \gamma \frac{\mathrm{c}(A)}{\log(1/\mathrm{c}(A))}$.*

It is an open problem to close the gap between the bounds of Theorem 6 and Theorem 7. It is also an open problem to generalize Theorem 5 to higher dimension, e.g., by giving a lower bound on $\mathrm{c}(A)$ in terms of $\mathrm{b}_{d-1}(A)$ for sets $A \subseteq \mathbb{R}^d$ satisfying some suitable topological restrictions.

Bibliography

[1] E. Ackerman: On the maximum number of edges in topological graphs with no four pairwise crossing edges. *Discrete & Computational Geometry*, 41(3):365–375, 2009.

[2] P. K. Agarwal, B. Aronov, J. Pach, R. Pollack, and M. Sharir: Quasi-planar graphs have a linear number of edges. *Combinatorica*, 17(1):1–9, 1997.

[3] O. Aichholzer, S. Cabello, R. Fabila-Monroy, D. Flores-Peñaloza, T. Hackl, C. Huemer, F. Hurtado, and D. Wood: Edge-removal and non-crossing configurations in geometric graphs. *Discrete Mathematics and Theoretical Computer Science*, 12(1), 2010.

[4] P. Angelini, G. Di Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter: Testing planarity of partially embedded graphs. *ACM Transactions on Algorithms*, 11(4):32:1–32:42, 2015.

[5] M. Balko, V. Jelínek, P. Valtr, and B. Walczak: On the Beer index of convexity and its variants. In L. Arge and J. Pach, editors, *International Symposium on Computational Geometry*, volume 34 of *Leibniz International Proceedings in Informatics*, pages 406–420, 2015.

[6] G. Beer: Continuity properties of the visibility function. *Michigan Mathematical Journal*, 20:297–302, 1973.

[7] G. Beer: The index of convexity and the visibility function. *Pacific Journal of Mathematics*, 44(1):59–67, 1973.

[8] G. Beer: The index of convexity and parallel bodies. *Pacific Journal of Mathematics*, 53(2):337–345, 1974.

[9] T. Biedl, M. Kaufmann, and P. Mutzel: Drawing planar partitions II: HH-drawings. In J. Hromkovič and O. Sýkora, editors, *Graph-Theoretic Concepts in Computer Science*, volume 1517 of *Lecture Notes in Computer Science*, pages 124–136, 1998.

[10] T. Bläsius and I. Rutter: Simultaneous PQ-ordering with applications to constrained embedding problems. In *Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1030–1043, 2013.

[11] T. Bläsius and I. Rutter: A new perspective on clustered planarity as a combinatorial embedding problem. In C. Duncan and A. Symvonis, editors, *Graph Drawing*, volume 8871 of *Lecture Notes in Computer Science*, pages 440–451, 2014.

[12] K. S. Booth and G. S. Lueker: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335 – 379, 1976.

[13] J. M. Boyer and W. J. Myrvold: On the cutting edge: Simplified $O(n)$ planarity by edge addition. *Journal of Graph Algorithms and Applications*, 8(3):241–273, 2004.

[14] S. Cabello, J. Cibulka, J. Kynčl, M. Saumell, and P. Valtr: Peeling potatoes near-optimally in near-linear time. In *Symposium on Computational Geometry*, SOCG'14, pages 224–231. ACM, 2014.

[15] M. Chimani, G. Di Battista, F. Frati, and K. Klein: Advances on testing c-planarity of embedded flat clustered graphs. In C. Duncan and A. Symvonis, editors, *Graph Drawing*, volume 8871 of *Lecture Notes in Computer Science*, pages 416–427, 2014.

[16] J. Černý, Z. Dvořák, V. Jelínek, and J. Kára: Noncrossing Hamiltonian paths in geometric graphs. *Discrete Applied Mathematics*, 155(9):1096–1105, 2007.

[17] G. Da Lozzo, V. Jelínek, J. Kratochvíl, and I. Rutter: Planar embeddings with small and uniform faces. In H.-K. Ahn and C.-S. Shin, editors, *Algorithms and Computation*, volume 8889 of *Lecture Notes in Computer Science*, pages 633–645, 2014.

[18] H. de Fraysseix, P. Ossona de Mendez, and P. Rosenstiehl: Trémaux trees and planarity. *International Journal of Foundations of Computer Science*, 17(05):1017–1029, 2006.

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

[20] G. Di Battista and R. Tamassia: Incremental planarity testing. In *30th Annual Symposium on Foundations of Computer Science*, pages 436–441, 1989.

[21] V. Dujmović, D. Eppstein, M. Suderman, and D. R. Wood: Drawings of planar graphs with few slopes and segments. *Computational Geometry*, 38(3):194 – 212, 2007.

[22] V. Dujmović, M. Suderman, and D. R. Wood: Really straight graph drawings. In J. Pach, editor, *Graph Drawing*, volume 3383 of *Lecture Notes in Computer Science*, pages 122–132, 2005.

[23] P. Erdős, R. L. Graham, P. Montgomery, B. L. Rothschild, J. Spencer, and E. G. Straus: Euclidean Ramsey theorems I. *Journal of Combinatorial Theory, Series A*, 14(3):341–363, 1973.

[24] P. Erdős, R. L. Graham, P. Montgomery, B. L. Rothschild, J. Spencer, and E. G. Straus: Euclidean Ramsey theorems II. In *Infinite and Finite Sets Colloq., Keszthely*, volume 10 of *Colloq. Math. Soc. János Bolyai*, pages 529–557, 1973.

[25] P. Erdős, R. L. Graham, P. Montgomery, B. L. Rothschild, J. Spencer, and E. G. Straus: Euclidean Ramsey theorems III. In *Infinite and Finite Sets Colloq., Keszthely*, volume 10 of *Colloq. Math. Soc. János Bolyai*, pages 559–583, 1973.

[26] I. Fáry: On straight-line representation of planar graphs. *Acta Sci. Math. (Szeged)*, 11, 1948.

[27] Q.-W. Feng, R. F. Cohen, and P. Eades: Planarity for clustered graphs. In P. Spirakis, editor, *Algorithms — ESA '95*, volume 979 of *Lecture Notes in Computer Science*, pages 213–226, 1995.

[28] R. Fulek: Towards the Hanani-Tutte theorem for clustered graphs. In D. Kratsch and I. Todinca, editors, *Graph-Theoretic Concepts in Computer Science*, pages 176–188, 2014.

[29] R. Fulek, J. Kynčl, I. Malinović, and D. Pálvölgyi: Clustered planarity testing revisited. In C. Duncan and A. Symvonis, editors, *Graph Drawing*, pages 428–439, 2014.

[30] M. T. Goodrich, G. S. Lueker, and J. Z. Sun: C-planarity of extrovert clustered graphs. In P. Healy and N. S. Nikolov, editors, *Graph Drawing*, volume 3843 of *Lecture Notes in Computer Science*, pages 211–222, 2006.

[31] C. Gutwenger, M. Jünger, S. Leipert, P. Mutzel, M. Percan, and R. Weiskircher: Advances in c-planarity testing of clustered graphs. In M. T. Goodrich and S. G. Kobourov, editors, *Graph Drawing*, volume 2528 of *Lecture Notes in Computer Science*, pages 220–236, 2002.

[32] C. Gutwenger and P. Mutzel: A linear time implementation of SPQR-trees. In J. Marks, editor, *Graph Drawing*, volume 1984 of *LNCS*, pages 77–90, 2000.

[33] J. Hopcroft and R. Tarjan: Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974.

[34] V. Jelínek, J. Kratochvíl, and I. Rutter: A Kuratowski-type theorem for planarity of partially embedded graphs. *Computational Geometry*, 46(4):466–492, 2013.

[35] V. Jelínek, E. Jelínková, J. Kratochvíl, and B. Lidický: Clustered planarity: Embedded clustered graphs with two-component clusters. In I. G. Tollis and M. Patrignani, editors, *Graph Drawing*, volume 5417 of *Lecture Notes in Computer Science*, pages 121–132, 2009.

[36] V. Jelínek, E. Jelínková, J. Kratochvíl, B. Lidický, M. Tesař, and T. Vyskočil: The planar slope number of planar partial 3-trees of bounded degree. *Graphs and Combinatorics*, 29(4):981–1005, 2013.

[37] V. Jelínek, J. Kynčl, R. Stolař, and T. Valla: Monochromatic triangles in two-colored plane. *Combinatorica*, 29(6):699–718, 2009.

[38] V. Jelínek, O. Suchý, M. Tesař, and T. Vyskočil: Clustered planarity: Clusters with few outgoing edges. In I. G. Tollis and M. Patrignani, editors, *Graph Drawing*, volume 5417 of *Lecture Notes in Computer Science*, pages 102–113, 2009.

[39] E. Jelínková, J. Kára, J. Kratochvíl, M. Pergel, O. Suchý, and T. Vyskočil: Clustered planarity: Small clusters in cycles and Eulerian graphs. *Journal of Graph Algorithms and Applications*, 13(3):379–422, 2009.

[40] G. Károlyi, J. Pach, and G. Tóth: Ramsey-type results for geometric graphs, I. *Discrete & Computational Geometry*, 18(3):247–255, 1997.

[41] B. Keszegh, J. Pach, and D. Pálvölgyi: Drawing planar graphs of bounded degree with few slopes. In U. Brandes and S. Cornelsen, editors, *Graph Drawing*, volume 6502 of *Lecture Notes in Computer Science*, pages 293–304. Springer Berlin Heidelberg, 2011.

[42] K. Knauer, P. Micek, and B. Walczak: Outerplanar graph drawings with few slopes. *Computational Geometry: Theory and Applications*, 47(5):614–624, 2014.

[43] K. Kuratowski: Sur le problème des courbes gauches en topologie. *Fund. Math.*, 15:271–283, 1930.

[44] T. Lengauer: Hierarchical planarity testing algorithms. *Journal of the ACM*, 36(3):474–509, July 1989.

[45] P. Mutzel and R. Weiskircher: Optimizing over all combinatorial embeddings of a planar graph. In *Integer Programming and Combinatorial Optimization*, volume 1610 of *Lecture Notes in Computer Science*, pages 361–376, 1999.

[46] J. Pach and J. Törőcsik: Some geometric applications of Dilworth's theorem. *Discrete & Computational Geometry*, 12(1):1–7, 1994.

[47] J. Pach and G. Tóth: Monotone drawings of planar graphs. *Journal of Graph Theory*, 46(1):39–47, 2004.

[48] M. Patrignani: On extending a partial straight-line drawing. *International Journal of Foundations of Computer Science*, 17(05):1061–1069, 2006.

[49] N. Robertson and P.D. Seymour: Graph minors XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65 – 110, 1995.

[50] G. Rote: The degree of convexity. In *Abstracts of the 29th European Workshop on Computational Geometry*, pages 69–72, 2013.

[51] M. Schaefer: Toward a theory of planarity: Hanani-Tutte and planarity variants. *Journal of Graph Algorithms and Applications*, 17(4):367–440, 2013.

[52] M. Schaefer: Picking planar edges; or, drawing a graph with a planar subgraph. In C. Duncan and A. Symvonis, editors, *Graph Drawing*, volume 8871 of *Lecture Notes in Computer Science* pages 13–24, 2014.

[53] L. E. Shader: All right triangles are Ramsey in E^2! *Journal of Combinatorial Theory, Series A*, 20(3):385–389, 1976.

[54] H. I. Stern: Polygonal entropy: a convexity measure for polygons. *Pattern Recognition Letters*, 10(4):229–235, 1989.

[55] W.T. Tutte: Toward a theory of crossing numbers. *Journal of Combinatorial Theory*, 8(1):45 – 53, 1970.

[56] P. Valtr: On geometric graphs with no k pairwise parallel edges. *Discrete & Computational Geometry*, 19(3):461–469, 1998.

[57] K. Wagner: Bemerkungen zum Vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 36:26–32, 1936.

[58] K. Wagner: Über eine Eigenschaft der ebenen Komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.

[59] G. J. Woeginger: Embeddings of planar graphs that minimize the number of long-face cycles. *Operations Research Letters*, 30(3):167–168, 2002.

Appendix A

Non-crossing Hamiltonian Paths

Available online at www.sciencedirect.com
ScienceDirect
Discrete Applied Mathematics 155 (2007) 1096 – 1105
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Noncrossing Hamiltonian paths in geometric graphs

Jakub Černý, Zdeněk Dvořák, Vít Jelínek, Jan Kára

Department of Applied Mathematics, Charles University, Malostranské náměstí 25, 118 00 Praha 1, Czech Republic

Received 8 January 2004; received in revised form 23 December 2004; accepted 22 December 2005
Available online 2 January 2007

Abstract

A geometric graph is a graph embedded in the plane in such a way that vertices correspond to points in general position and edges correspond to segments connecting the appropriate points. A noncrossing Hamiltonian path in a geometric graph is a Hamiltonian path which does not contain any intersecting pair of edges. In the paper, we study a problem asked by Micha Perles: determine the largest number $h(n)$ such that when we remove any set of $h(n)$ edges from any complete geometric graph on n vertices, the resulting graph still has a noncrossing Hamiltonian path. We prove that $h(n) \geqslant (1/2\sqrt{2})\sqrt{n}$. We also establish several results related to special classes of geometric graphs. Let $h_1(n)$ denote the largest number such that when we remove edges of an arbitrary complete subgraph of size at most $h_1(n)$ from a complete geometric graph on n vertices the resulting graph still has a noncrossing Hamiltonian path. We prove that $\frac{1}{\sqrt{2}}\sqrt{n} < h_1(n) < 3\sqrt{n}$. Let $h_2(n)$ denote the largest number such that when we remove an arbitrary star with at most $h_2(n)$ edges from a complete geometric graph on n vertices the resulting graph still has a noncrossing Hamiltonian path. We show that $h_2(n) = \lceil n/2 \rceil - 1$. Further we prove that when we remove any matching from a complete geometric graph the resulting graph will have a noncrossing Hamiltonian path.

Keywords: Geometric graph; Hamiltonian path

1. Introduction

A *geometric graph* is a graph drawn in the plane so that its vertices are represented by points in general position (i.e., there are no three collinear points) and its edges are straight-line segments connecting the corresponding vertices.

Lately geometric graphs have been intensively studied. There are many papers studying the smallest number of edges needed to guarantee the occurrence of some fixed subconfiguration in any geometric graph (the best known result of this type following from Euler's polyhedral formula is that any geometric graph with at least $3n - 5$ edges must have two edges which intersect). Interesting results of this sort may be found in [2,4,7–10] to name a few or in the surveys on geometric graphs [5,6].

In our paper we study the existence of a noncrossing Hamiltonian path (i.e. Hamiltonian path which does not cross itself) in a given geometric graph. In particular we concentrate on a problem presented by Micha Perles on DIMACS Workshop on Geometric Graph Theory in 2002, which asks to determine the largest possible number $h(n)$ such that every geometric graph on n vertices with at least $\binom{n}{2} - h(n)$ edges has a noncrossing Hamiltonian path. The same

E-mail addresses: kuba@kam.mff.cuni.cz (J. Černý), rakdver@kam.mff.cuni.cz (Z. Dvořák), jelinek@kam.mff.cuni.cz (V. Jelínek), kara@kam.mff.cuni.cz (J. Kára).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2005.12.010

question for noncrossing Hamiltonian cycle is uninteresting. If the points are in convex position and we delete an edge on the boundary then there is no noncrossing Hamiltonian cycle anymore. Perles himself has shown the upper bound $h(n) < n/2$ and the lower bound $h(n) \geqslant 2$. Apart from improving the lower bound of $h(n)$, we also focus on the restriction of the problem to some classes of geometric graphs. Let $h_1(n)$ denote the largest number such that when we remove the edges of an arbitrary complete subgraph of size at most $h_1(n)$ from a complete geometric graph on n vertices, the resulting graph always has a noncrossing Hamiltonian path. We prove that $\frac{1}{\sqrt{2}}\sqrt{n} < h_1(n) < 3\sqrt{n}$ (Theorems 1 and 3). Let $h_2(n)$ denote the largest number such that when we remove an arbitrary star with at most $h_2(n)$ edges from a complete geometric graph on n vertices, the resulting graph still has a noncrossing Hamiltonian path. In Theorem 6 we show that $h_2(n) = \lceil n/2 \rceil - 1$. We also prove that $h_3(n) = \lceil n/2 \rceil - 1$, where $h_3(n)$ is the largest number such that when we remove at most $h_3(n)$ arbitrary edges from the complete geometric graph on n vertices in convex position, then the graph still has a noncrossing Hamiltonian path (Theorems 4 and 5). The previous equality also follows from an unpublished result of Micha Perles, which has been proved independently to our research. Further we prove that when we remove any matching from acomplete geometric graph, the resulting graph has a noncrossing Hamiltonian path (Theorem 2).

The paper is organized as follows: In Section 2 we introduce basic definitions and notation. In Sections 3 and 4 we study complete geometric graphs with removed complete subgraph and we prove the asymptotically tight bounds on the size of complete subgraph removed. In Section 5 we prove the tight bounds on the number of edges removed from convex geometric graph and in Section 6 we prove the tight bounds on the size of a star removed from a complete geometric graph.

The paper is based on the paper from Graph Drawing 2003 [3]. We added Section 7 with open problems.

2. Definitions and notation

In this section we introduce basic definitions and notation used throughout this paper. A *geometric graph* G is an ordered pair (V, E) where V is a set of points in general position in the plane (called *vertices of* G) and E is a set of straight–line segments connecting two vertices (called *edges of* G). A *Hamiltonian path* in a graph G is a path contained in G which visits all the vertices of G. A *noncrossing Hamiltonian path* in a geometric graph $G = (V, E)$ is a Hamiltonian path which does not intersect itself. A *convex hull* of a set of points $X \subset R^2$, $X = \{x_1, \ldots, x_n\}$ is a set of points $H = \{h \in R^2 : \exists a_1, \ldots a_n$ such that $\forall i \in \{1, \ldots, n\}$ $a_i \in R, a_i \geqslant 0, \sum_{i=1}^n a_i = 1$ and $h = \sum_{i=1}^n a_i x_i\}$. We say that a point p lies *below a line* l (line l must not be parallel to y-axis) if it lies in the half-plane defined by l which contains $-\infty$ on the y axis. Similarly we use the term *above a line*. A point u lies *to the left of* v if the x-coordinate of u is less than or equal to the x-coordinate of v. Similarly we define that u is *to the right of* v. Let s be a segment in R^2 defined by two endpoints $u, v \in R^2$, where u is to the left of v. We say that a point $p \in R^2$ lies *below the segment* s if it lies below a line defined by u and v, to the right of u and to the left of v. Let $Z \subset R^2$, $Z = \{z_1, \ldots, z_n\}$ be a set of points in the plane. An *x-monotone order* of Z is an ordering of Z in which the x-coordinates of the points form a monotone sequence. Analogously we define a *y-monotone order* of Z. A point $z \in Z$ is called an *extremal point* of Z if it belongs to the boundary of the convex hull of Z. A segment uv, with u and v in Z, is called an *extremal segment* of Z if it is a subset of the boundary of the convex hull of Z.

3. The lower bound for complements of cliques

In the following two sections we consider a particular class $\mathscr{C}$ of geometric graphs—the complements of complete subgraphs. A geometric graph $G = (V, E)$ is in $\mathscr{C}$ iff there exist $X, Y \subseteq V$ such that $V = X \cup Y$, $X \cap Y = \emptyset$ and E is the set of all the possible edges with at least one enpoint in Y (i.e. G is obtained from a complete graph by removing the edges of a complete subgraph). We prove that any geometric graph $G \in \mathscr{C}$ with $|X| \leqslant \frac{1}{\sqrt{2}}\sqrt{|V|}$ has a noncrossing Hamiltonian path.

Lemma 1. *Let $G = (V, E)$ be a geometric graph, $G \in \mathscr{C}$. If there exists a line l such that all the vertices of X are in one half-plane defined by l and at least $|X|$ vertices of Y are in the other half-plane, then there exists a noncrossing Hamiltonian path in G.*

Proof. We may WLOG assume that the line l is parallel to the y-axis and that all the vertices of the set X are in the left half-plane. The algorithm for producing a noncrossing Hamiltonian path in this configuration is shown on Fig. 1. We

```
Input: Two sets of points in the plane X, Y and line l
Output: Noncrossing Hamiltonian path
begin
  PathLen := 0;
  V := X ∪ Y;
  Left := False;
  LastLeft := True;
  forever do begin
    Seg := CrossSegment(V, l);
    if Seg = ∅ then
      break;
    if Left then begin
      if (Path[PathLen-1] ∈ Y) or (Seg.Left ∈ Y) then begin
        Path[PathLen] := Seg.Left;
        PathLen := PathLen + 1;
        V := V \ Seg.Left;
        LastLeft := True;
      end
      else
        Left := False;
    end
    else begin
      Path[PathLen] := Seg.Right;
      PathLen := PathLen + 1;
      V := V \ Seg.Right;
      Left := True;
      LastLeft := False;
    end;
  end;
  while V ≠ ∅ do begin
    if LastLeft then
      P := Leftmost(V)
    else
      P := Rightmost(V);
    Path[PathLen] := P;
    PathLen := PathLen + 1;
    V := V \ P;
  end;
  Output(Path)
end.
```

Fig. 1. An algorithm for finding a noncrossing Hamiltonian path. Function `CrossSegment` returns the extremal segment of the given set which intersects l (actually there are two such segments so the function returns the one which intersects l at position with bigger y coordinate—we suppose l is parallel to y axis). Functions `Leftmost` and `Rightmost` return, respectively, the leftmost and the rightmost point of the given set.

iterate the procedure described below to find a noncrossing Hamiltonian path in G. The procedure is a generalization of an algorithm developed by Abellanas et al. [1]. In each step, the procedure adds a new vertex to the path that it has constructed in the previous steps, according to the following rules: first, we take the upper extremal segment of the vertices not yet added to the path (the whole set V in the beginning) which crosses l (see Fig. 2 on the left). If the last vertex added to the path or the left endpoint of the extremal segment is from Y, then the left endpoint of the segment is added to the path (Fig. 2 on the right), otherwise we add the right endpoint (Fig. 3 on the left). If no vertices were added yet, we may add either of the endpoints. If l does not cross the convex hull, then we simply add the remaining vertices to the path in an x-monotone order. See Fig. 3 on the right for an example of the path produced by this algorithm.

It is clear that the algorithm finishes when it adds all the vertices to the path. It is also easy to see that there are no two consecutive vertices of X on the constructed path and hence it is indeed a path in G. The only place in the algorithm, where two vertices of X might be added consecutively to the path, is when the convex hull of the vertices not yet on the path no longer intersects l, all these remaining vertices lie to the left of l, and the algorithm is adding them in an x-monotone order. But at that time there is at most one vertex of X not on the path: the other vertices of X were added in the previous steps because there are at least $|X|$ vertices of Y in the right half-plane, and each of them has been added to the path immediately after a vertex of Y, with the possible exception of the first vertex of the path.

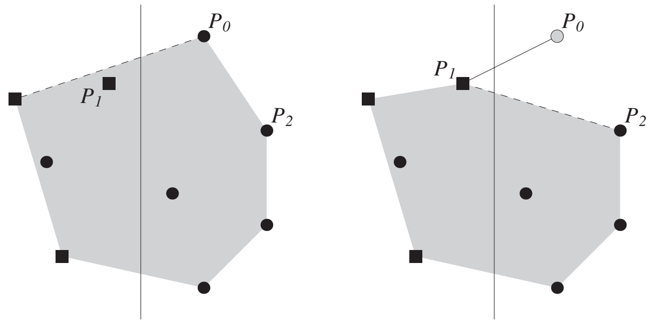


Fig. 2. The left figure shows the first step of the algorithm. The right figure shows the second step of the algorithm—add left end of the segment to the path.

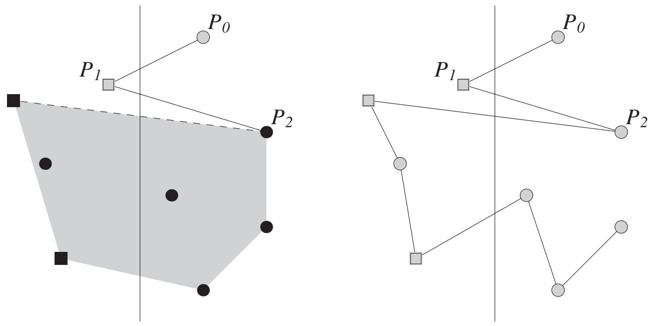


Fig. 3. On the left there is the third step of the algorithm—add right end of the segment to the path. On the right there is the whole noncrossing Hamiltonian path.

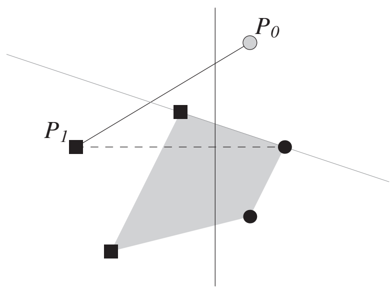


Fig. 4. If the segment to new vertex intersected the convex hull of the remaining vertices the previous vertex on the path would have to lie in the lower half-plane.

It remains to prove that the path is noncrossing. We check that after each step of the algorithm, the path does not intersect the convex hull of the remaining vertices including the vertex just added to the path. From this it is obvious that the path does not intersect itself (at each vertex each of the following edges of the path must lie in the convex hull and the previous edges lie outside of it). When the path contains only one vertex, the claim is obviously true. When we add a new vertex to the path, the edge connecting the new vertex to the previous vertex on the path cannot intersect the convex hull of the remaining vertices—if the edge intersected the convex hull, then the previous vertex on the path would have to lie in the lower half-plane defined by the upper extremal segment of the remaining vertices intersecting l (see Fig. 4). But then we get contradiction with the choice of the vertex in the previous step of the algorithm (the vertex cannot be an endpoint of the upper extremal segment intersecting l). From the induction we know that no other

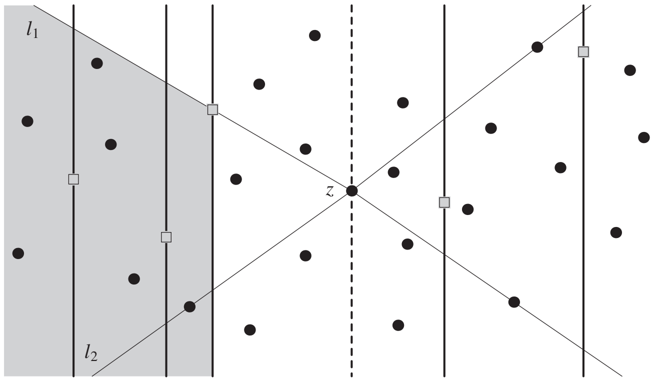


Fig. 5. Partitioning of the plane into the strips and choice of the vertices to which Lemma 1 should be applied.

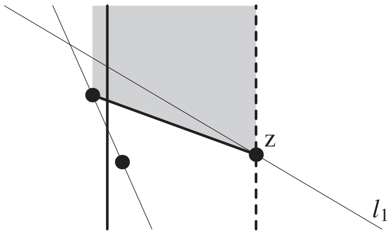


Fig. 6. There are no vertices lying above the first segment of the path and below the line l_1.

edge of the path can intersect the convex hull of the remaining vertices and so we have proven that the path does not intersect itself. □

Now we prove a similar result for a general choice of the sets X and Y:

Theorem 1. *Let $G=(V,E)$ be a geometric graph, $G \in \mathscr{C}$. If $|Y| \geqslant 2|X|(|X|+1)$ then there exists a noncrossing Hamiltonian path in G.*

Proof. Assume WLOG that all vertices of X have different x coordinates. Partition the plane into $|X|+1$ vertical strips separated by vertical lines passing through the points of X (see Fig. 5). It follows from the pigeonhole principle that there is a strip S with at least $2 \cdot |X|$ vertices of Y in it. Let x_l denote the number of vertices of X to the left of S (including the vertex on the left boundary of S). Similarly let x_r denote the number of vertices of X to the right of S. Now we can certainly choose a vertex z of S such that there are at least $2x_l$ vertices in S to the left from z and at least $2x_r$ vertices in S to the right from z. The vertical line passing through z splits S into two strips, denoted S_l (the left one) and S_r (the right one).

We now describe a procedure to find a noncrossing path starting in z and containing all vertices to the left of z. We find lines l_1, l_2 such that $z \in l_1 \cap l_2$, both l_1 and l_2 contain some vertex lying to the left from S_l and there is no vertex lying to the left from S_l which would lie above l_1 or below l_2 (see Fig. 5). It is clear that in S_l there are either at least x_l vertices below l_1 or at least x_l vertices above l_2. Assume WLOG that there are at least x_l vertices below l_1. Let Z denote the set of vertices lying to the left from z and below l_1, including the two vertices on the line l_1. Now we can apply Lemma 1 to the set Z (line l from the statement of the lemma is the left boundary of the strip S). From the lemma we get a noncrossing path starting in z containing all the vertices of Z (note that z is an endpoint of the upper extremal segment of Z intersecting l). Because there are no vertices of Z above the first segment of the path (all the vertices must lie below a line defined by the extremal segment from the second step of the procedure from Lemma 1 and the whole area above the first segment of the path lies above this line—see Fig. 6), we can replace the first segment of the path by the path going through all the vertices in S_l above the line l_1 in an x-monotone order (see Fig. 7). By the replacing we got a

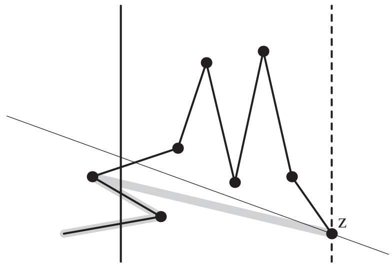


Fig. 7. Replacing the first segment of the path with an x-monotone path.

noncrossing path starting in z and using all the vertices of V to the left of z. Similarly we can get a path starting from z and containing all the vertices of V to the right of z, then join these two paths in z and get a noncrossing Hamiltonian path for G.

It remains to count the constant:

$$n = |X| + |Y| \geqslant |X| + 2|X|(|X| + 1) \geqslant 2|X|^2$$

So we get $|X| \leqslant \frac{1}{\sqrt{2}}\sqrt{n}$. $\square$

Now using Theorem 1 we can prove the following corollary:

Corollary 1. *Let $G = (V, E)$ be a geometric graph and let $k = \left(\binom{|V|}{2} - |E|\right)$ (i.e., k is the number of edges not present in G). If $8k^2 + 6k \leqslant |V|$ then there exists a noncrossing Hamiltonian path in G.*

Proof. The idea of the proof is easy. We apply Theorem 1 for the original graph where X is the set of all vertices of the missing edges.

More formally, let X be the set of all vertices of V with degree less than $|V| - 1$. The size of X is clearly less than or equal to $2k$. Let $Y = V \backslash X$. From the statement of the corollary we know that $|Y| \geqslant 2|X|(|X| + 1)$ and hence the assumptions from the statement of Theorem 1 are satisfied and we can conclude that G has a noncrossing Hamiltonian path. This gives a lower bound $h(n) \geqslant (1/2\sqrt{2})\sqrt{n}$ defined in the introduction. $\square$

Using the algorithm with a similar idea as the algorithm in Lemma 1 we can also prove the following result for the complements of matchings:

Theorem 2. *Let $G = (V, E)$ be a geometric graph which is a complement of a matching (i.e. a graph with minimum degree $|V| - 2$) and $|V| \geqslant 3$. Then G has a noncrossing Hamiltonian path.*

Proof. The case when $|V| = 3$ is trivial so we can assume $|V| \geqslant 4$. We use the following algorithm for a construction of a noncrossing Hamiltonian path: first take any extremal point of V to be the first vertex of the path. Let x be the last vertex on the path constructed so far. Then at each step we choose a vertex y which is an extremal point of the remaining vertices such that $\{x, y\}$ is an edge in G which does not intersect the convex hull of the remaining points, and we add y to the path. The vertex y with the desired properties always exists if there are at least two remaining vertices. Let H' denote the set of points outside the path in the previous step of the algorithm and H the set of points outside the path in the current step. In the previous step, x was an extremal vertex of H' and so in the current situation there must be some segment y_1y_2 which is an extremal segment of H, such that neither of segments xy_1 and xy_2 do intersect the convex hull of H. Because at least one of these segments must be an edge in G (its complement was a matching) we have just proven the existence of y.

If there is only one remaining vertex and it is not connected by an edge to the last point on the path, we cannot finish the path. Let y_1 be the last remaining vertex and y_2, y_3, and y_4 be the last vertices on the path in the reverse order (remember that $|V| \geqslant 4$). In this situation we remove the vertices y_2 and y_3 from the path to get the situation from

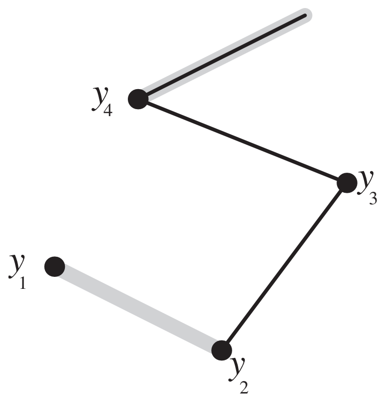


Fig. 8. The situation when we cannot add the last point to the noncrossing path.

Fig. 8. Because y_1 and y_2 are not connected by an edge of G, we know that y_4 must be connected by an edge to both y_1 and y_2 and one of these edges does not intersect the convex hull of the remaining points. So we can WLOG add y_1 to the path and then finish the path by adding y_3 and y_2. $\square$

4. The upper bound

In this section we prove that there exist geometric graphs in $\mathscr{C}$ such that the size of X is $O(\sqrt{|V|})$ and the graphs do not have a noncrossing Hamiltonian path. By proving this we get two asymptotically tight bounds on the function $h_1(n)$ defined in the introduction.

Definition 1. Let V be a set of points in convex position. We say that a point $u \in V$ is next to a point $v \in V$, if the segment uv is an extremal segment of V.

Lemma 2. *Let $G = (V, E)$ be a geometric graph such that all the vertices of V are in convex position. Let $P = (p_1, \dots, p_n)$ be a noncrossing Hamiltonian path in G. Then for any $j \in \{2, \dots, n\}$ the vertex p_j is next to a vertex p_k for some $k < j$. In particular, for any $j \in \{1, \dots, n-2\}$ it holds that among the three vertices p_j, p_{j+1}, p_{j+2} at least two are next to each other.*

Proof. Assume that the statement does not hold. Let p_j be the first vertex among $p_2, p_3, \dots, p_n$ that is not next to any of the previous vertices of P. Let p_r and p_s be the two vertices next to p_j, where $j < r < s$. The path $Q=(p_j, p_{j+1}, \dots, p_n)$ contains both p_r and p_s, hence Q intersects the edge $p_{j-1}p_j$. However, P contains the edge $p_{j-1}p_j$ as well as all the edges of Q, so P is not a noncrossing path, contrary to our assumptions. $\square$

Theorem 3. *For each $n_0 \in N$ there exists $n \in N$, $n_0 \leqslant n$ such that there is a geometric graph $G = (V, E)$, $G \in \mathscr{C}$, $|V| = n$ satisfying $|X| < 3\sqrt{n}$ without a noncrossing Hamiltonian path.*

Proof. Fix some n_0. Let n be the smallest natural number greater than n_0 which is the square of some natural number. Now we describe a geometric graph on n vertices with the desired properties. We place n vertices of the graph on a circle. Then we split the vertices into $\sqrt{n}$ groups (each of size $\sqrt{n}$) in such a way that each group forms a contiguous sequence on the circle. Now we define the partitioning of V into X and Y (and by this we determine the edges of the graph). We choose arbitrarily one group (let us call it g) and put all its vertices to X. In the other groups put the first and last vertices to X and the remaining points to Y (see Fig. 9). Clearly $|X| = (\sqrt{n} - 1)2 + \sqrt{n} < 3\sqrt{n}$, so it only remains to prove that the graph does not have a noncrossing Hamiltonian path.

Consider the first vertex u in g. Because $u \in X$, it must be connected to some vertex v of Y by an edge of the path. We write g' for the group containing the vertex v. Because both neighbors of u on the convex hull are also from X, v cannot be a neighbor of u. From this we can also trivially conclude that neither u nor v can be the endpoints of the Hamiltonian path. Now we focus our attention on the part of the path which should contain all the remaining vertices

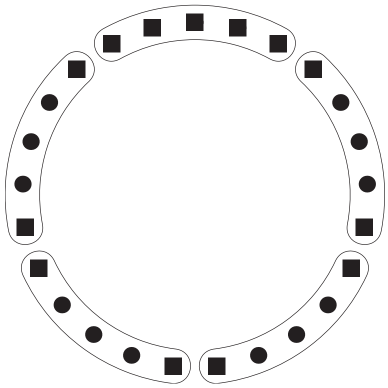


Fig. 9. Construction of a graph on n vertices with $|X| = O(\sqrt{n})$ without a noncrossing Hamiltonian path.

of g. From Lemma 2 we know that the path must go from v either to some other vertex of g' or to the neighbor of u in g. From any vertex of g the path must return back to g' to the neighbor of the vertex last used in g. From this we get that on the noncrossing Hamiltonian path there must be an alternation of vertices from g and from g' in such a way that between any two vertices from g there must be a vertex from $g' \cap Y$. But we have $\sqrt{n}$ vertices from g and only $\sqrt{n} - 2$ vertices from $g' \cap Y$ and so this is impossible. □

5. Vertices in convex position

In the following section we consider a class $\mathscr{D}$ of convex geometric graphs. A geometric graph $G = (V, E)$ is in $\mathscr{D}$ iff the vertices of G are in a convex position. We show that if we remove $\lceil |V|/2 \rceil - 1$ edges from the complete geometric graph then the noncrossing Hamiltonian path still exists (Theorem 4) but if we remove $\lceil |V|/2 \rceil$ edges it need not exist (Theorem 5). Note that the bounds are tight.

Theorem 4. *Let $G = (V, E)$ be a geometric graph, $G \in \mathscr{D}$, $n = |V|$. Let $\overline{G} = (V, F)$ be the complement of G. If $|F| \leqslant \lceil n/2 \rceil - 1$ then there exists a noncrossing Hamiltonian path in G.*

Proof. Let $v_0, v_1, ..., v_{n-1}$ be the vertices of G in clockwise order, starting with an arbitrary one. Consider the complete geometric graph $G' = (V, E \cup F)$. Let P_i be the path $v_i, v_{i+1}, v_{i-1}, v_{i+2}, v_{i-2}, \ldots$ (counting the indices modulo n) in G'. We observe that the paths $P_1, \ldots, P_{\lfloor n/2 \rfloor}$ are pairwise disjoint noncrossing Hamiltonian paths in G'. Since $|F| \leqslant \lceil n/2 \rceil - 1$ we are done for n even—at least one of the paths must avoid F and hence it is a noncrossing Hamiltonian path in G. If n is odd we observe that there are $\lfloor n/2 \rfloor$ edges $\{v_0, v_{n-1}\}, \{v_1, v_{n-2}\}, \ldots, \{v_{\lfloor n/2 \rfloor - 1}, v_{\lfloor n/2 \rfloor + 1}\}$ which are not in any P_i. Let A denote this set of edges. WLOG the set V forms the vertex set of a regular convex n-gon. Observe that every edge of G' can be mapped to an edge of A by rotating G' along its centre of rotational symmetry. So we can WLOG assume that at least one of the edges of F is in A (and hence it is not in any of the paths P_i). Now we can conclude using the same argument as for n even that one of the P_is is a noncrossing Hamiltonian path in G. □

Theorem 5. *For each n, $n \geqslant 2$ there exists a geometric graph $G_n = (V_n, E_n)$ such that $G_n \in \mathscr{D}$, $|V_n| = n$, $|E_n| = \binom{n}{2} - \lceil n/2 \rceil$ and G_n does not have a noncrossing Hamiltonian path.*

Proof. Let $v_0, v_1, \ldots, v_{n-1}$ be the vertices of G_n in clockwise order, starting with an arbitrary one. First we make an easy observation: the first (and the last) edge of a noncrossing Hamiltonian path P is an extremal segment of V. Consequently, if $v_i v_j$ is an edge of such a path but not an extremal segment, then P contains at least one extremal segment from each of the intervals $v_i \ldots v_j$ and $v_j \ldots v_i$.

Let $k = \lceil n/2 \rceil$. We choose $F_n = \{\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}\}$ and E_n as the complement of F_n. Let $B = \{v_0, \ldots, v_k\}$. Suppose there exists a noncrossing Hamiltonian path P avoiding F_n. No edge in P may join two points of B, as then by the observation above it would have to contain an edge of F_n. Therefore, B is an independent set of P, which is impossible as the largest independent set of P is of size k. □

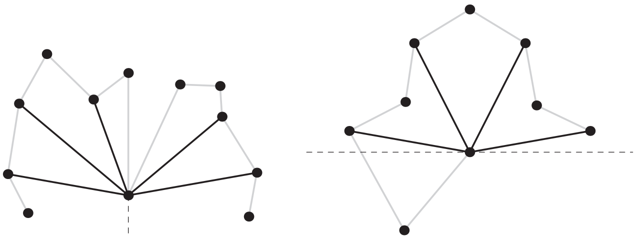


Fig. 10. Construction of a noncrossing Hamiltonian path. There are either two uncovered vertices in one of the cones (on the left) or every cone contains exactly one uncovered vertex (on the right).

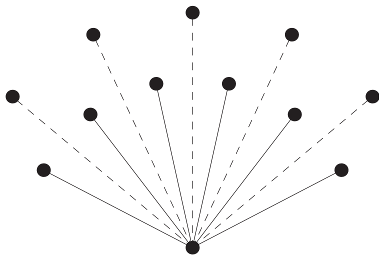


Fig. 11. Complement of a star without a noncrossing Hamiltonian path.

6. Complement of star

In this section we consider a class of geometric graphs $\mathscr{S}$. A geometric graph $G=(V,E)$ is in $\mathscr{S}$ iff E is a complement of F, where F is the edge set of a star $K_{1,k}$. We prove that for $k \leqslant \lceil |V|/2 \rceil - 1$ there always exists a noncrossing Hamiltonian path but for $k \geqslant \lceil |V|/2 \rceil$ it need not exist.

Theorem 6. *For any geometric graph $G=(V,E)$ on n vertices, $G \in \mathscr{S}$ such that $|F| \leqslant \lceil n/2 \rceil - 1$ there exists a noncrossing Hamiltonian path in G. For any n, $n \geqslant 2$ there exists a geometric graph on n vertices G_n, $G_n \in \mathscr{S}$ with $|F_n| = \lceil n/2 \rceil$ such that there is no noncrossing Hamiltonian path in G_n.*

Proof. Let C be the center of a star F, where F has at most $\lceil n/2 \rceil - 1$ edges. We partition the plane into cones by extending the edges of F into rays starting in C. If there is a cone that contains at least two vertices that are not covered by F, we use the construction from Fig. 10 (left). Otherwise there must be exactly one vertex in each of the cones. At most one of the cones spans an angle greater than straight, let x be the vertex inside this cone if such a cone exists, otherwise let x be an arbitrary vertex not covered by F. The half-line starting in C passing through x splits its cone into two, so at least one of them spans an angle smaller than straight. Now we use the construction from Fig. 10 (right).

We proceed to prove the second claim of the theorem. Let V_n and F_n look as in Fig. 11. Note that the boundary of the convex hull of V_n contains the vertex C, two edges of F_n, and all the vertices not covered by F_n. Moreover, every cone defined by the rays extending the edges of F_n contains at most one vertex not covered by F_n.

Suppose there is a noncrossing Hamiltonian path P. If C is the first vertex of P, then the second vertex of P is one of the uncovered ones, the third one belongs to one of the half-planes determined by the first edge and P cannot get to the other half-plane without intersecting its first edge, so the path cannot be Hamiltonian. Similarly, if C is an interior vertex of P, the edges of P adjacent to it split the remaining vertices into three nonempty parts, and P cannot cover more than two of them without intersecting itself. $\square$

7. Conclusion and open problems

In the previous sections we have shown the bounds $h(n) = \Omega(\sqrt{n})$ and $h(n) = O(n)$ for the function h defined in the introduction. We have also shown linear lower bounds for the restrictions of h on some particular classes of geometric graphs. This leads us to conjecture that the linear upper bound of h is asymptotically tight, i.e. we have $h(n) = \theta(n)$.

The conjecture above is related to the following open problem: determine the values of the function $f(n)$ defined as the maximum number f such that every complete geometric graph on n vertices contains a collection of f pairwise edge-disjoint noncrossing Hamiltonian paths. Clearly $f(n) \leqslant h(n)+1$. Moreover, we can observe that the two functions are related by means of integer programming duality. Define $\mathscr{P}(G) = \{P \subseteq E(G);\ P$ is the edge set of a noncrossing Hamiltonian path in $G\}$.

Let G be a given complete geometric graph. Consider the following integer programming problem, with integer-valued variables $x_e;\ e \in E(G)$:

$$\begin{aligned} & H(G) = \min \sum_{e \in E(G)} x_e \\ \text{s.t.}\quad & \forall e \in E(G), \quad x_e \geqslant 0, \\ & \forall P \in \mathscr{P}(G), \quad \sum_{e \in P} x_e \geqslant 1. \end{aligned}$$

In other words, $H(G)$ is the smallest cardinality of a set $R \subseteq E(G)$ such that the graph $G' = (V, E \backslash R)$ has no noncrossing Hamiltonian path. It follows that $h(n)+1 = \min\{H(G);\ \ G$ is a complete geometric graph on n vertices$\}$. If we consider the dual of the problem above, we obtain the following problem, with integer-valued variables $y_P;\ \ P \in \mathscr{P}$:

$$\begin{aligned} & F(G) = \max \sum_{P \in \mathscr{P}(G)} y_P \\ \text{s.t.}\quad & \forall P \in \mathscr{P}(G), \quad y_P \geqslant 0, \\ & \forall e \in E(G), \quad \sum_{P:\ e \in P} y_P \leqslant 1. \end{aligned}$$

Observe that $F(G)$ is the maximum size of a collection of pairwise edge-disjoint noncrossing Hamiltonian paths in G, so we have $f(n) = \min\{F(G);\ \ G$ is a complete geometric graph on n vertices$\}$.

It is clear that $F(G) \leqslant H(G)$ for every complete geometric graph G and it is also clear that $f(n) \leqslant h(n)+1$. However, it is an open problem to establish the exact behavior of f and h, or to decide whether these inequalities are asymptotically tight.

Acknowledgments

We would like to thank Jan Kratochvíl and Pavel Valtr who brought the problem to our attention, discussed it with us and helped us with the preparation of this paper. We would also like to thank János Pach for pointing out relevant references.

References

[1] M. Abellanas, J. García, G. Hernández, M. Noy, P. Ramos, Bipartite embeddings of trees in the plane, Discrete Appl. Math. 93 (1999) 141–148.
[2] P.K. Agarwal, B. Aronov, J. Pach, R. Pollack, M. Sharir, Quasi-planar graphs have a linear number of edges, Combinatorica 17 (1997) 1–9.
[3] J. Černý, Z. Dvořák, V. Jelínek, J. Kára, Noncrossing Hamiltonian paths in geometric graphs, Graph Drawing 2003 (Perugia), Lecture Notes in Computer Science, vol. 2912, 2004, pp. 86–97.
[4] A. Marcus, G. Tardos, On topological graphs without self-intersecting 4-cycles, in preparation.
[5] J. Pach, Geometric graph theory, Surveys in combinatorics, Canterbury, 1999, pp. 167–200, London Mathematical Society Lecture Note Series, vol. 267, Cambridge University Press, Cambridge, 1999.
[6] J. Pach, P.K. Agarwal, Combinatorial Geometry, Wiley Interscience, New York, 1995.
[7] J. Pach, R. Radoičić, G. Tardos, G. Tóth, Geometric graphs with no self–intersecting path of length three, Graph Drawing, Lecture Notes in Computer Science, vol. 2528, Springer, Berlin, 2002, pp. 295–311.
[8] R. Pinchasi, R. Radoičić, On the number of edges in geometric graphs with no selfintersecting cycle of length 4, Proceedings of the 19th Annual Symposium on Computational Geometry, 2003, pp. 98–103.
[9] G. Tardos, On the number of edges in a geometric graph with no short self-intersecting paths, in preparation.
[10] P. Valtr, Graph drawings with no k pairwise crossing edges, Graph Drawing (Rome), Lecture Notes in Computer Science, vol. 1353, 1997, pp. 205–218.

Appendix B

Embedded Clustered Graphs with Two-Component Clusters

Clustered Planarity: Embedded Clustered Graphs with Two-Component Clusters
(Extended Abstract)

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

⋆ 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.

I.G. Tollis and M. Patrignani (Eds.): GD 2008, LNCS 5417, pp. 121–132, 2009.

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: 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 this extended abstract, we present a simplified version of the algorithm which assumes that the cluster hierarchy is flat. We also omit some of the proofs.

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*[1]. 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 proofs are omitted due to space constraints.

[1] 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

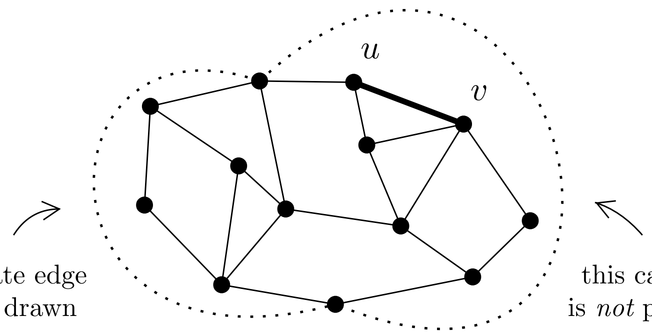


Fig. 1. Two candidate edges connecting the same pair of vertices in the outer face

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.

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.

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.

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 are omitted from this extended abstract.

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.

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.

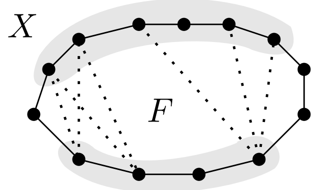


Fig. 2. A face F with two bundles of X-edges

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

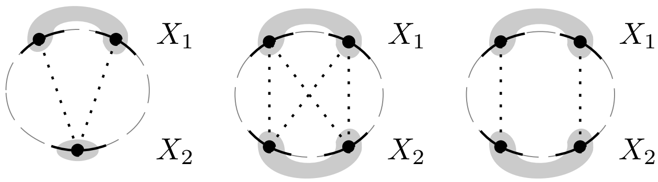


Fig. 3. Possible configurations of permitted edges of an F-cluster X

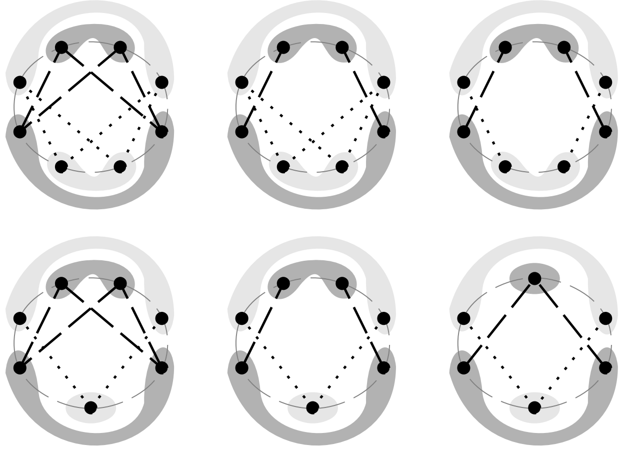


Fig. 4. Mutual positions of permitted edges of two crossing F-clusters

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).

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:

– 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, vol. 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 9(3), 391–413 (2005); special issue In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 100–110. Springer, Heidelberg (2005)
3. Dahlhaus, E.: A linear time algorithm to recognize clustered planar graphs and its parallelization. In: Lucchesi, C.L., Moura, A.V. (eds.) 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.G. (ed.) ESA 1995. LNCS, vol. 979, pp. 213–226. Springer, Heidelberg (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, vol. 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: Goodrich, M.T., Kobourov, S.G. (eds.) GD 2002. LNCS, vol. 2528, pp. 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, pp. 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)

Appendix C

Clusters with Few Outgoing Edges

Clustered Planarity: Clusters with Few Outgoing Edges

Vít Jelínek*, Ondřej Suchý*, Marek Tesař, and Tomáš Vyskočil*

Department of Applied Mathematics
Charles University
Malostranské nám. 25, 118 00 Praha, Czech Republic
{jelinek,suchy,tesulo,tiger}@kam.mff.cuni.cz

Abstract. We present a linear algorithm for c-planarity testing of clustered graphs, in which every cluster has at most four outgoing edges.

1 Introduction

Clustered planarity is one of the challenges of contemporary Graph Drawing. It arises naturally when we want to draw the graph with further constraints on embedding of the vertices. This includes for example visualizing a computer network with the computers of the same department, faculty and institution being grouped together. Another application is in designing an integrated circuit with the connectors of each components being close to each other and the logical parts of the circuit being grouped together. There are many other applications including visualizations of process interaction, social networks etc.

The concept of the clustered graph—a graph equipped with a system of subsets of vertices (called clusters), that can be recursive— was first introduced by Feng et al. in [7]. In the same paper they also proved that clustered planarity (shortly c-planarity) can be tested in polynomial time for c-connected clustered graphs (where each cluster induces a connected subgraph of the underlying graph). This was later improved by Dahlhaus [4] to a linear time algorithm. The paper [7] also contains a useful characterization of the c-planar graphs: Graph is c-planar if and only if there is a set of edges (usually called a saturator) that can be added to this graph to obtain a c-connected c-planar clustered graph.

Since then many algorithms for testing the c-planarity were based on searching for a saturator. These include an $O(n^2)$-time algorithm for "almost" c-connected clustered graphs by Gutwenger et al. in [9,10]. An efficient algorithm for clusters with cyclic structure on a cycle was developed in [3]. The case of disjoint clusters on an embedded graph with small faces was recently addressed in [5]. Very similar result was at the same time independently published by Jelínková et al. [12]. The paper [12] also contains an $O(n^3)$-time algorithm for clusters of size at most three on a rib-Eulerian graph. This is an Eulerian graph that is obtained from a constant size 3-connected graph by multiplying and then subdividing edges.

* Supported by grant 201/05/H014 of the Czech Science Foundation.

I.G. Tollis and M. Patrignani (Eds.): GD 2008, LNCS 5417, pp. 102–113, 2009.

Another approach is to mimic the original proof of Feng et al. [7] where the behavior of the connected clusters is described by special trees. In this way a slight generalization to extrovert clustered graphs was given by Goodrich et al. [8]. In an extrovert clustered graph the parent cluster of any disconnected cluster is connected and every component of any disconnected cluster is incident to an edge which leads outside of its parent cluster.

We should also mention that every c-planar graph can be drawn by straight lines with clusters represented by convex polygons [6]. Another interesting contribution is the characterization of completely connected clustered graphs (where each subgraph induced by a cluster and its complement are connected) [1]: A completely connected clustered graph is c-planar if and only if the underlying graph is planar. More results on c-planarity can be found in [2]. Despite the number of results the complexity of testing the c-planarity for general instances remains open.

In this paper we focus on the situation where the number of outgoing edges of each cluster is small. We notice that in this case the behavior of the clusters can be simulated by special graphs, no matter whether the subgraph induced by the cluster is connected or not. We use these ideas to develop a linear time algorithm to test such graphs for c-planarity. As far as we know this is the first algorithm that can be used in the cases where the underlying graph is not connected at all or has very few edges in total. In particular we prove the following theorem:

Theorem 1. *Clustered planarity can be decided in linear time for instances, where each cluster has at most 4 outgoing edges.*

Section 2 is devoted to the basic definitions. We also show there that if there is a cluster with no outgoing edges, then the instance could be split into an instance formed by the subclusters of the cluster and one formed by the rest. In Section 3 we show how to replace the clusters by special graphs with the same behavior and prove that this does not affect the c-planarity. The algorithm is described in Section 4, together with the proofs of the correctness and the running time. In Section 5 we show that the approach cannot be generalized this way to the case of five or more outgoing edges.

2 Preliminaries

Let S_r denote the set of all permutations of the set $\{1, 2, \ldots, r\}$. A permutation $\pi \in S_r$ is represented by r-tuple $(\pi(1) \ldots \pi(r))$.

Regarding the graph notations, we follow the standard notation on finite loopless graphs. A graph is an ordered pair $G = (V, E)$, where V is the set of vertices and E is the set of edges i.e. pairs of vertices. We simply write uv instead of $\{u, v\}$ for edges. If $U \subseteq V$, then $G[U]$ is the induced subgraph of G on vertices U and $G \setminus U = G[V \setminus U]$. Let n denote the number of vertices $|V|$ of the graph G.

A *cluster set* on the graph $G = (V, E)$ is a set $\mathcal{C} \subseteq \mathcal{P}(V(G))$ such that for all $C, D \in \mathcal{C}$, either C and D are disjoint or they are in inclusion; the pair $(G, \mathcal{C})$ is called a *clustered graph*. The elements of $\mathcal{C}$ are called *clusters*. A *clustered* *planar embedding* of $(G, \mathcal{C})$ is a planar embedding *emb* of G together with a mapping emb_c that assigns to every cluster $C \in \mathcal{C}$ a planar region $emb_c(C)$ whose boundary is a closed Jordan curve and such that

- for each vertex $v \in V$ and every cluster $C \in \mathcal{C}$, it holds that $emb(v) \in emb_c(C)$ if and only if $v \in C$,
- for every two clusters C and D, the regions $emb_c(C)$ and $emb_c(D)$ are disjoint (in inclusion) if and only if C and D are disjoint (in inclusion, respectively), and
- for every edge $e \in E$ and every cluster $C \in \mathcal{C}$ the curve $emb(e)$ crosses the boundary of $emb_c(C)$ at most once.

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

The following observation is a trivial consequence of the definition:

Remark 1. A pair $(G, \emptyset)$ is c-planar if and only if the graph G is planar.

We say that $C \in \mathcal{C}$ is a *cluster of the bottom-most level* if there is no $C' \in \mathcal{C}$ such that $C' \subset C$.

An edge $e = uv$ is an *outgoing edge* of a cluster C if $u \in C, v \in V \setminus C$ or vice versa.[1] Let $r(C) = |\{e = uv | e \in E, u \in C, v \in V \setminus C\}|$ denote the number of outgoing edges of a cluster C. If the cluster is clear from context we will just use notation r instead of $r(C)$.

Lemma 1. *If C has no outgoing edges then $(G, \mathcal{C})$ is c-planar if and only if $(G \setminus C, \mathcal{C}_1)$ and $(G[C], \mathcal{C}_2)$ are c-planar, where $\mathcal{C}_1 = \{A \setminus C | A \in \mathcal{C}, A \neq C, A \supset C\} \cup \{A | A \in \mathcal{C}, A \cap C = \emptyset\}$ and $\mathcal{C}_2 = \{B | B \in \mathcal{C}, B \neq C, B \subset C\}$.*

Proof. The direction from left to right is easy, we just omit from the embedding the parts that are no longer necessary.

So suppose that we have a c-planar embedding emb_1 of $(G \setminus C, \mathcal{C}_1)$ and a c-planar embedding emb_2 of $(G[C], \mathcal{C}_2)$. Take an arbitrary point x in the plane, such that for all clusters $A \in \mathcal{C}_1$ the following holds: x lies inside the region $(emb_1)_c(A)$ if and only if $C \subseteq A$. Suppose that there is neither vertex nor edge of $G \setminus C$ nor border of a cluster of $\mathcal{C}_1$ in distance less than ϵ from x in emb_1. Now shrink the embedding emb_2 so that it fits into the $\frac{\epsilon}{2}$-disc centered in x. Then take this disc as the embedding of C.

It is easy to check that we obtain a c-planar embedding of $(G, \mathcal{C})$, since the embeddings emb_1 and emb_2 cross neither each other nor the embedding of C, the inclusions of the clusters are preserved and the embedding of the cluster C contains exactly the embedding of the vertices, edges and clusters it should contain. □

[1] Such an edge is called *edge incident with C* in [1,3,7,9] and *extrovert* edge in [8].

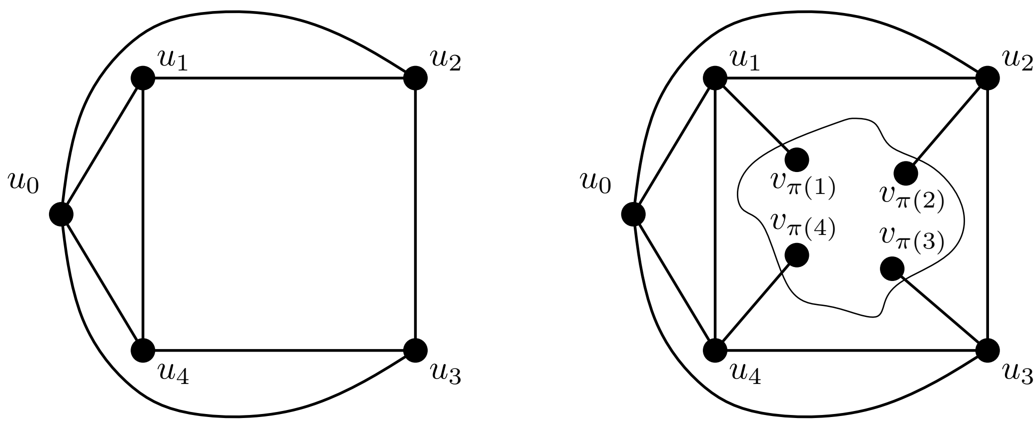


Fig. 1. The test graph T and the graph T_C^π from Definition 1

3 Replacement of Clusters by Graphs

Through this section we suppose, that we have some fixed cluster $C \in \mathcal{C}$ of bottom-most level, that has at most 4 outgoing edges. Having Lemma 1 in hand we assume that $1 \le r = r(C) \le 4$.

We denote the outgoing edges by $\{e_1, \ldots, e_r\}$. We also suppose that $e_i = v_i w_i$ for all i, where $v_i \in C$ and $w_i \in V \setminus C$ (maybe $w_i = w_j$ or $v_i = v_j$ for some $i \neq j$).

We denote by T the following *test graph* $T = (\{u_0, u_1, u_2, u_3, u_4\}, \{u_0u_1, u_0u_2, u_0u_3, u_0u_4, u_1u_2, u_2u_3, u_3u_4, u_4u_1\})$ (see Fig. 1).

Definition 1. *We say that the cluster C* admits a permutation $\pi \in S_r$ *if and only if the graph T_C^π created from $T \cup G[C]$ by adding edges $u_i v_{\pi(i)}, 1 \le i \le r$ is planar.*

Lemma 2. *If the cluster C admits the permutation $\pi \in S_r$ then there exists a planar embedding of the graph T_C^π such that the vertices of C are embedded inside and the vertex u_0 outside the cycle $u_1, \ldots, u_4, u_1$ of T. Moreover we can prescribe this cycle to be oriented clockwise in the embedding.*

Proof. First we take some planar embedding of the graph T_C^π. Now we take the edges incident with u_0 in the clockwise order $u_0u_{i_1}, u_0u_{i_2}, u_0u_{i_3}, u_0u_{i_4}$. For every u_0u_i and u_0u_j two consecutive of them (either $\{i, j\} = \{i_k, i_{k+1}\}$ for some $k = 1, 2$ or 3 or $\{i, j\} = \{i_1, i_4\}$) we can draw a new curve from u_i to u_j along the curve u_iu_0 and then u_0u_j so that it does not cross any other edge and area surrounded by the curves u_iu_0, u_0u_j and the new curve contains no vertex (see Fig. 2).

Suppose for a contradiction that some of the newly drawn curves connects two non-adjacent vertices, for example u_1 and u_3 (the case of u_2 and u_4 being similar). Since the new curves connect u_1 to at most one of the vertices u_2 and u_4 and we drew two curves from each u_i, we also connected u_2 and u_4. But this means that the newly drawn curves together with the original edges form

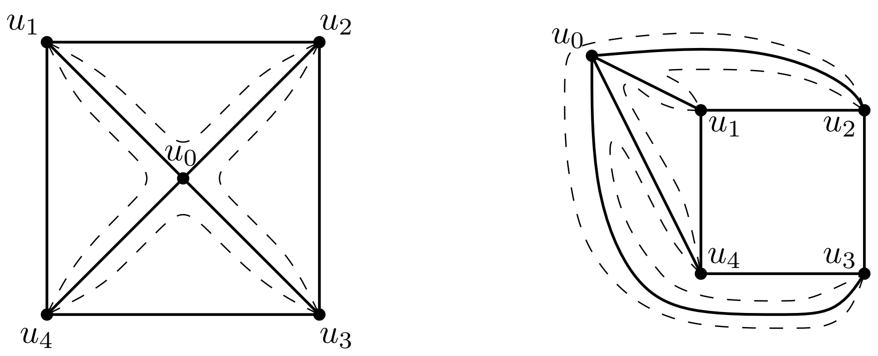


Fig. 2. Situation from the proof of Lemma 2

a planar embedding of K_5, which is a contradiction. So we know that all the curves that we drew newly connect two already adjacent vertices of the cycle.

Now we take these newly drawn curves as the embedding of the edges of the cycle. Then there is just u_0 inside the cycle and it remains to change the outer face to one of the newly obtained empty triangles, such that the vertex u_0 will be on the boundary of the outer face.

If the cycle is embedded in wrong direction, then we take the axis symmetry of the embedding. $\square$

Lemma 3. *If* $(G, \mathcal{C})$ *is c-planar, then* C *admits some permutation.*

Proof. We suppose that $(G, \mathcal{C})$ is c-planar and we fix a planar embedding *emb*. Let f be the boundary of $emb_C(C)$ (so f is a closed Jordan curve). Now we can start in an arbitrary point of this curve and move along this curve in the clockwise direction and we cross the edges $e_1, e_2, \ldots, e_r$ in some order $e_{i_1}, e_{i_2}, \ldots, e_{i_r}$. Denote the crossing points as $P_1, P_2, \ldots, P_r$ (in the same order). If $r < 4$ then we can choose new points $P_{r+1}, \ldots, P_4$ in such a way, that we meet the points $P_1, \ldots, P_4$ in this order when we move along the curve f in the clockwise direction and all these points are distinct.

Now we consider the planar embedding emb' of $G[C]$ which corresponds to the embedding *emb* of the graph G, place new vertices $u_1, \ldots, u_4$ to the points $P_1, \ldots, P_4$ and a vertex u_0 outside of the region bounded by the curve f. Clearly we can add edges $(u_1, v_{i_1}), \ldots, (u_r, v_{i_r})$ and embed these edges on curves which corresponded to edges $e_1, \ldots, e_r$ inside of the region $emb_C(C)$ and we can also add edges $(u_1, u_2), (u_2, u_3), (u_3, u_4)$ and (u_4, u_1) and embed them on the curve f in such a way that these edges may intersect only in vertices u_1, u_2, u_3 or u_4. It is clear that we can add edges $(u_0, u_1), \ldots, (u_0, u_4)$ and embed them in such a way that these edges will be outside of the region bounded by f and every two edges will cross only in the vertex u_0.

This way we obtain a planar embedding of the graph T_C^π where $\pi = (a_{i_1} \ldots a_{i_r})$. Thus C admits the permutation π. $\square$

Lemma 4. *If the cluster* C *admits a permutation* $\pi = (a_1 a_2 \ldots a_r)$ *then it also admits permutations* $(a_r a_1 \ldots a_{r-1})$ *and* $(a_r a_{r-1} \ldots a_1)$*.*

Proof. We obtain the planar embedding of $T_C^\delta, \delta = (a_r a_1 \dots a_{r-1})$ from the planar embedding of T_C^π simply by relabeling the vertices such that u_1 becomes u_2, u_2 becomes u_3, u_3 becomes u_4 and u_4 becomes u_1 and if $r < 4$ then it is necessary to replace the edge $v_{a_r} u_{r+1}$ by a new edge $v_{a_r} u_1$ which goes along the edges $v_{a_r} u_{r+1}, u_{r+1} u_{r+2}, \dots, u_4 u_1$ such that it doesn't cross any other edge.

For $r \geq 3$ the second part can be done similarly – it is enough to relabel such that u_1 becomes u_3 and u_3 becomes u_1 and if $r = 4$ then we use the first part to achieve permutation $(a_r a_{r-1} \dots a_1)$. For $r < 3$ the first part also proves the second part. □

We can now define a relation $\sim'$ on the permutations from the set S_r by $(a_1 a_2 \dots a_r) \sim' (a_r a_1 \dots a_{r-1})$ and $(a_1 a_2 \dots a_r) \sim' (a_r a_{r-1} \dots a_1)$. If we take $\sim$ to be the transitive closure of $\sim'$, then it is easy to show that $\sim$ is also reflexive and symetric. Thus $\sim$ is an equivalence. We will sometimes call the equivalence classes of this equivalence *circular permutations* The sets S_1, S_2, S_3 have just one equivalence class under $\sim$ while the set S_4 is partitioned into following three equivalence classes (they can be distinguished by the number that is "opposite" to the number 1):

$$S_4^2 = \{(1324), (3241), (2413), (4132), (4231), (1423), (3142), (2314)\},$$

$$S_4^3 = \{(1234), (2341), (3412), (4123), (4321), (1432), (2143), (3214)\},$$

$$S_4^4 = \{(1243), (2431), (4312), (3124), (3421), (1342), (2134), (4213)\}.$$

Definition 2. *We define the* corresponding graph for cluster C *as follows (see Fig 3).*

1. *If $r \leq 3$ and C admits some permutation, then the corresponding graph for C is R_r.*
2. *If there is a labeling of the outgoing edges such that C admits permutations from S_4^2, S_4^3, S_4^4 then the corresponding graph for C with this labeling is R_4^{234}.*
3. *If there is a labeling of the outgoing edges such that C admits a permutation from S_4^2 and from S_4^3, but no permutation from S_4^4 then the corresponding graph for C with this labeling is R_4^{23}.*
4. *If there is a labeling of the outgoing edges such that C admits a permutation from S_4^2, but no permutation from $S_4^3 \cup S_4^4$ then the corresponding graph for C with this labeling is R_4^2.*

Clearly, if $r \leq 3$ then the cluster C has unique corresponding graph. Since the sets S_4^2, S_4^3, and S_4^4 form a decomposition of S_4, from Lemma 4 we know that the cluster C admits all permutations from some non-empty combination of sets S_4^2, S_4^3, and S_4^4.

If the cluster C admits just permutations from the set S_4^i then by relabeling of edge e_2 by e_i and incident vertices v_2 by v_i and w_2 by w_i (if $i = 2$ we don't need to do it) we get labeling of the cluster C which admits only permutations from the set S_4^2. So the cluster C has unique corresponding graph.

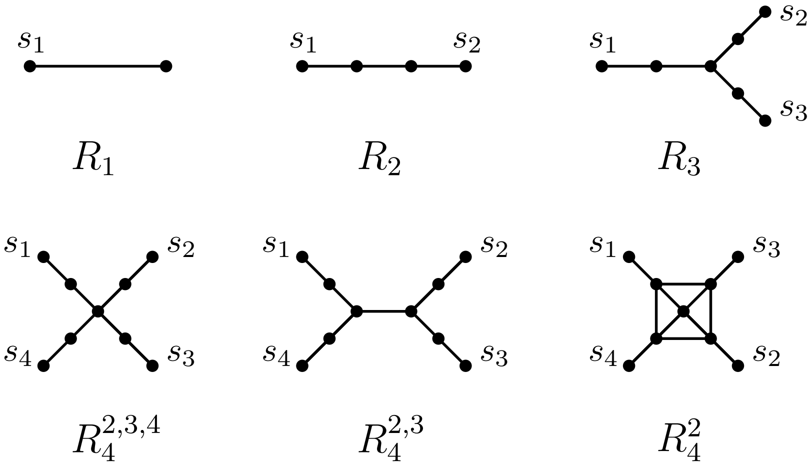


Fig. 3. The graphs $R_1, R_2, R_3, R_4^{234}, R_4^{23}$ and R_4^2

If the cluster C admits just permutations from two distinct sets S_4^i and S_4^j then we make similar relabeling of outgoing edges and incident vertices such that resulting relabeling makes the cluster C admit just permutations from the sets S_4^2 and S_4^3 and the cluster C has unique corresponding graph.

As a consequence we get the following corollary.

Corollary 1. *If C admits a permutation then there is a labeling of outgoing edges of C such that C has a corresponding graph with this labeling.*

For the rest of the paper we will use this new labeling.

Definition 3. *Let C be a cluster of the bottom-most level with outgoing edges $e_1, \ldots, e_r$ where $1 \leq r \leq 4$, $e_i = v_i w_i$ for all i, where $v_i \in C$ and $w_i \in V \setminus C$. Let R be a corresponding graph to the cluster C in this labeling. Then a* replacement of cluster C by a corresponding graph R *in $(G, \mathcal{C})$ is a clustered graph $(G', \mathcal{C}')$ such that G' is created from $(G \setminus C) \cup R$ by unification of $w_1, \ldots, w_r$ with $s_1, \ldots, s_r$ (respectively) and $\mathcal{C}'$ is created from $\mathcal{C} \setminus \{C\}$ by replacing every $C' \supseteq C$ by $(C' \setminus C) \cup (V(R) \setminus \{s_1, \ldots, s_r\})$.*

Proposition 1. *Let $(G', \mathcal{C}')$ be the replacement of cluster C by a corresponding graph R. Then $(G, \mathcal{C})$ is c-planar if and only if $(G', \mathcal{C}')$ is c-planar.*

Proof. ("$\Rightarrow$":) We suppose that $(G, \mathcal{C})$ is c-planar and we fix some planar embedding emb. Without loss of generality we can suppose that $emb_{\mathcal{C}}(C)$ is a disc (because this region is homeomorphic to a disc). Suppose that the edges $e_1, \ldots, e_r$ cross the boundary of $emb_{\mathcal{C}}(C)$ in (clockwise) order $e_{i_1}, \ldots, e_{i_r}$ and without loss of generality $i_1 = 1$.

If $r < 4$ then we simply remove cluster C with edges $e_1, \ldots, e_r$ and draw the graph R_r corresponding to C in a such way, that we identify vertex s_i with w_i for all $i \in \{1, \ldots, r\}$ and all other vertices of R_r draw inside $emb_{\mathcal{C}}(C)$ in such a way, that edges of R_r don't cross any other edge of original graph nor other edge of R_r. This is clearly possible, it is enough to draw the edges outside the

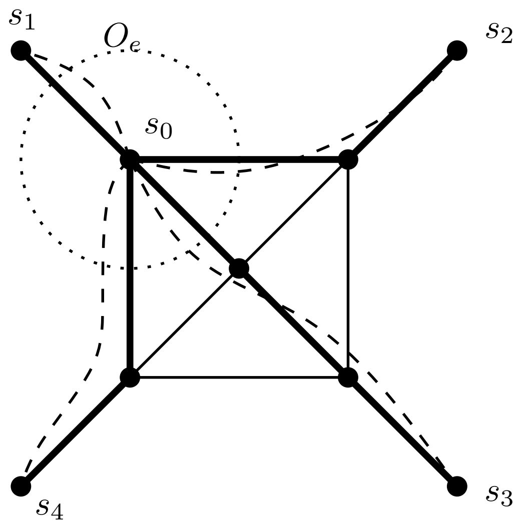


Fig. 4. Situation from the proof of Proposition 1(part "$\Leftarrow$")

disc $emb_{\mathcal{C}}(C)$ along the deleted edges $e_1, \ldots, e_r$ and inside $emb_{\mathcal{C}}(C)$ we can draw edges (or parts of edges) as noncrossing segments. This embedding of G' shows that $(G', \mathcal{C}')$ is c-planar.

If the corresponding graph for C is R_4^{234} then we can construct a c-planar embedding of C' in the same way as for $r < 4$.

If the corresponding graph for C is R_4^2 then the ordered set (i_1, i_2, i_3, i_4) must be equal to $(1, 3, 2, 4)$ or $(1, 4, 2, 3)$ because C admits only permutations from S_4^2 (otherwise we could find a permutation $\pi \notin S_4^2$ such that T_C^π is planar which is a contradiction). Now we delete the cluster C and add the graph R_4^2 in such a way that all the vertices of R_4^2 will be inside the disc $emb_{\mathcal{C}}(C)$ and we identify vertices s_i with w_i for all $i \in \{1, \ldots, 4\}$ and any edge of R_4^2 will not cross any original edge nor any new edge of R_4^2. This is also clearly possible, it is enough to draw the edges outside the disc $emb_{\mathcal{C}}(C)$ along the deleted edges $e_1, \ldots, e_4$ and inside $emb_{\mathcal{C}}(C)$ we can draw the edges (or parts of the edges) as noncrossing segments. This embedding of G' shows that $(G', \mathcal{C}')$ is c-planar.

If the corresponding graph for cluster C is R_4^{23} then we continue similarly as in the previous cases. The ordered set (i_1, i_2, i_3, i_4) must be equal to $(1, 3, 2, 4)$, $(1, 4, 2, 3)$, $(1, 2, 3, 4)$ or $(1, 4, 3, 2)$ so again it is easy to replace the vertices and the edges of C by the graph R_4^{23} by identifying the vertices s_i with w_i for all $i \in \{1, \ldots, 4\}$ which proves that $(G', \mathcal{C}')$ is c-planar again.

("$\Leftarrow$":) Suppose we have a c-planar embedding of $(G', \mathcal{C}')$. Moreover suppose that in the case $R = R_4^2$ there is nothing embedded in any interior face of R. This can be easily achieved in a similar way as in the proof of Lemma 2. We take an arbitrary spanning tree of the graph R and let s_0 denote its arbitrary vertex different from $s_1, \ldots, s_r$. Now draw the r curves connecting s_0 to $s_1, s_2, \ldots, s_r$ along the unique paths connecting the vertices in the tree, so that they do not cross each other nor anything in the embedding, except possibly for the edges of R. Then remove the original edges of R.

Now take some ϵ such that there are no edges, vertices nor clusters embedded in distance less than ϵ from s_0, except for the curves incident with s_0. Denote by O_ϵ the circle of radius ϵ with center s_0. Suppose P_i is the last intersection of the curve $s_0 s_i$ with O_ϵ. We can assume, that ϵ is so small, that if we label these curves clockwise $s_0 s_{i_1}, s_0 s_{i_2}, \ldots, s_0 s_{i_r}$ as they leave s_0, then $P_{i_1}, P_{i_2}, \ldots, P_{i_r}$ are the points P_i in the clockwise order along O_ϵ. (We can assume, that each curve in the embedding is formed by finitely many straight line segments and circular arcs.)

By case analysis we show, that C admits the permutation $\pi = (i_1 i_2 \ldots i_r)$. This is clear if $R = R_1, R_2, R_3$ or R_4^{234}. The graph $R_4^2 \backslash \{s_1, \ldots, s_4\}$ is 3-connected so the order of the edges is given in this case (up to the equivalence $\sim$) and the permutation π is in S_4^2. If $R = R_4^{23}$ and $\pi \in S_4^4$, then by connecting the neigbouring edges we obtain a planar embedding of $K_{3,3}$ — a contradiction.

So we take the planar embedding of T_C^π guaranteed by Lemma 2 and remove the vertex u_0. We can take a homeomorphic copy of this embedding of $T_C^\pi \setminus \{u_0\}$, in which the cycle u_1, u_2, u_3, u_4, u_1 coincides with a circle O_ϵ and the vertices $u_1, u_2, \ldots, u_r$ are embedded at the points $P_{i_1}, P_{i_2}, \ldots, P_{i_r}$, respectively. We replace the interior of O_ϵ by such an embedding.

We are ready to describe an embedding of $(G, \mathcal{C})$. For every i the concatenation of the curve $v_{\pi(i)} u_i = P_{\pi(i)}$ and $P_{\pi(i)} s_{\pi(i)}$ forms an embedding of the edge $v_{\pi(i)} w_{\pi(i)}$ that crosses no other edge of G' or $G[C]$. Moreover, it crosses the boundary of each cluster of C' at most once, since there were no cluster boundaries inside O_ϵ, curve $P_{\pi(i)} s_{\pi(i)}$ was drawn along some edges of R and among them only the one incident with $s_{\pi(i)}$ could cross some cluster boundary and also at most once, because we started with a c-planar embedding of $(G', \mathcal{C}')$. It remains to take O_ϵ as the boundary of the cluster C. It only crosses the edges $w_i v_i$. Furthermore, since curve $s_i P_i$ (recall that $s_i = w_i$) lies completely outside O_ϵ (except for P_i), while $P_i v_i$ lies completely inside O_ϵ (except for P_i), O_ϵ crosses the edge $w_i v_i$ exactly once (in the point P_i). There are no other crossings, since they would have to be in the original c-planar embedding of (G', C') too. $\square$

4 The Algorithm

The algorithm is described in Fig. 5.

Proposition 2. *The algorithm correctly decides c-planarity for instances, where each cluster has at most 4 outgoing edges.*

Proof. We first prove by the mathematical induction that for every $0 \le i \le |\mathcal{C}|$, the pair (G_i, C_i) is defined and c-planar if and only if $(G, \mathcal{C})$ is c-planar. This is certainly true for $i = 0$. Now suppose that this is true for every $i' < i$ and let us prove it for i.

In the case $r(C) = 0$ we have two possibilities. Either $G_{i-1}[C]$ is not planar, then also G is not planar and $(G_{i-1}, \mathcal{C}_{i-1})$ is definitely not c-planar. Then the algorithm correctly rejects (and $G_j, \mathcal{C}_j$ is not defined for $j \ge i$). Or $G_{i-1}[C]$ is planar and by Lemma 1 and Remark 1 pair $(G_{i-1}, \mathcal{C}_{i-1})$ is c-planar if and only

Input: Graph G and cluster set $\mathcal{C}$, where each cluster has at most 4 outgoing edges.
Task: Accept (G, C) if and only if (G, C) is clustered planar.

1. Set $G_0 := G, \mathcal{C}_0 := \mathcal{C}$.
2. For $i := 1$ to $|\mathcal{C}|$ do:
   (a) Let C be some cluster on the bottom-most level in $\mathcal{C}_{i-1}$.
   (b) If $r(C) = 0$ then
      i. If $G_{i-1}[C]$ is planar then set
         $G_i := G_{i-1} \setminus C$
         $\mathcal{C}_i := \{A \setminus C | A \in \mathcal{C}_{i-1} \setminus \{C\}, A \supseteq C\} \cup \{A | A \in \mathcal{C}_{i-1}, A \not\supseteq C\}$
      ii. else REJECT.
   (c) else
      i. For each permutation $\pi \in S_{r(C)}$ test whether C admits π (whether T_C^π is planar)
      ii. If C admits no permutation, then REJECT.
      iii. Let $(G_i, \mathcal{C}_i)$ be the replacement of cluster C by the corresponding graph in $(G_{i-1}, \mathcal{C}_{i-1})$.
3. If $G_{|\mathcal{C}|}$ is planar then ACCEPT, otherwise REJECT ($\mathcal{C}_{|\mathcal{C}|} = \emptyset$).

Fig. 5. An overview of the algorithm

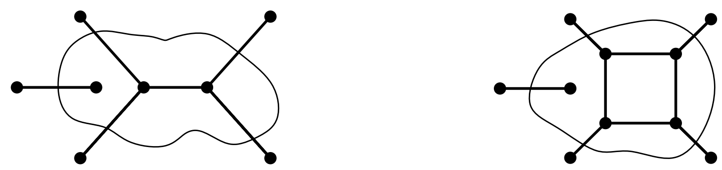
if $(G_i, \mathcal{C}_i)$ is, since $\{B | B \in \mathcal{C} \setminus \{C\}, B \subseteq C\}$ is empty (C is on the bottom-most level).

Now consider the case $1 \leq r(C) \leq 4$. If C admits no permutation, then by Lemma 3 the pair $(G_{i-1}, \mathcal{C}_{i-1})$ is not c-planar and the algorithm correctly rejects (and does not define $G_j, \mathcal{C}_j$ for $j \geq i$). Otherwise C has a corresponding graph by Corollary 1 and from the Proposition 1 we know that $(G_{i-1}, \mathcal{C}_{i-1})$ is c-planar if and only if $(G_i, \mathcal{C}_i)$ is c-planar.

Since $|\mathcal{C}_i| = |\mathcal{C}_{i-1}| - 1$ whenever defined, we have $|\mathcal{C}_{|\mathcal{C}|}| = 0$ and thus $\mathcal{C}_{|\mathcal{C}|} = \emptyset$ if $\mathcal{C}_{|\mathcal{C}|}$ is defined. But then $(G_{|\mathcal{C}|}, \mathcal{C}_{|\mathcal{C}|}) = (G_{|\mathcal{C}|}, \emptyset)$ is c-planar if and only if $G_{|\mathcal{C}|}$ is planar due to Remark 1, which completes the proof. □

Proposition 3. *The algorithm works in time $O(n)$.*

Proof. The cycle is executed at most $|\mathcal{C}|$ times, in each time we delete one cluster or reject. When we omit a planarity testing, complexity of each step of cycle in the algorithm is bounded by constant. We add constant number of vertices and if we have a suitable representation of clusters (for example tree representation) we can find cluster on the bottom-most level in constant time too. And then for these operations we need $|\mathcal{C}|$ in complexity time. The algorithm touches each vertex at most three times, when we add, test, and remove it. For vertices which we added later we paid before, by constant in each iteration. And for the original vertices we need extra n for planarity testing. Each vertex from the original graph we touch only once, because if we touch it we remove it or reject whole graph. Since $|\mathcal{C}|$ is bounded by $O(n)$, the complexity of our algorithm is $O(n)$. □



D_1 D_2

Fig. 6. Two clusters with 5 outgoing edges that cannot be represented by any connected graph

5 The Limits of the Approach

Let us consider clusters with more than 4 outgoing edges. Definition 1, Lemmas 2 and 4 easily generalize to this case as well as Lemma 3. The problem with the generalization is that there are disconnected clusters with 5 outgoing edges that admit a combination of permutations which cannot be represented by a connected graph. In particular it can be shown that the two clusters from Fig. 6 have this property.

Let us try to formalize the result. Consider a graph R that is supposed to be corresponding to some cluster C. Hence it has some distinguished vertices $s_1, \ldots, s_r$ of degree 1 that are supposed to be identified with the vertices of $G \setminus C$ when the cluster C is replaced by R in a graph G. Let $R' = V(R) \setminus \{s_1, \ldots, s_r\}$. We say that the graph R admits a permutation π, if the cluster R' of the clustered graph $(R, \{R'\})$ admits a permutation π.

Proposition 4. *There is no connected graph that admits the same number of permutations as the cluster D_1 from Fig. 6.*

Proof. We will count the circular permutations. In total there are 12 circular permutations on 5 elements, each representing 10 (standard) permutations. Observe first that D_1 admits 8 circular permutations. Now assume for a contradiction that there is a connected graph R (with distinguished vertices $s_1, \ldots, s_5$) that admits also 8 circular permutations.

We observe that whenever we take a subgraph Q of the graph R, $s_1, \ldots, s_5 \in V(Q)$, then the graph Q admits at least the same number of permutations, since we can just ommit the unnecessary parts from the appropriate embedding. Now consider a subtree T of R with leaves of T being exactly the vertices $s_1, \ldots, s_5$. It is clear that R has such a subgraph since R is connected.

Since T has 5 leaves, it has at most 3 vertices of degree at least 3 — either it has 3 vertices of degree 3, or one of degree 3 and one of degree 4, or just one of degree 5. It is not hard to check that in the first two cases T admits 4 and 6 circular permutations, respectively. Thus in this cases R cannot admit 8 circular permutations. The tree with just one vertex of degree 5 (among the vertices of degree at least 3) admits all 12 circular permutations. Thus we know that T must be some subdivision of $K_{1,5}$.

If R contains no path connecting two different branches of T, then clearly R admits the same permutations as T i.e. all 12 circular permutations. On the other hand, if R contains a path between two branches of T, then there is another tree T', subgraph of R, that has one vertex of degree 4 and one vertex of degree 3. But this means that R admits at most 6 permutations — a contradiction. □

References

1. Cornelsen, S., Wagner, D.: Completely connected clustered graphs. Journal of Discrete Algorithms 4(2), 313–323 (2006)
2. Cortese, P.F., Di Battista, G.: Clustered planarity. In: ACM SoCG 2005, pp. 32–34 (2005)
3. Cortese, P.F., Di Battista, G., Patrignani, M., Pizzonia, M.: Clustering cycles into cycles of clusters. Journal of Graph Algorithms and Applications 9(3), 391–413 (2005); In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 391–413. Springer, Heidelberg (2005)
4. Dahlhaus, E.: A linear time algorithm to recognize clustered planar graphs and its parallelization. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN 1998. LNCS, vol. 1380, pp. 239–248. Springer, Heidelberg (1998)
5. 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)
6. Eades, P., Feng, Q.W., Lin, X., Nagamochi, H.: Straight-line drawing algorithms for hierarchical graphs and clustered Graphs. Algorithmica 44, 1–32 (2006)
7. Feng, Q.W., Cohen, R.F., Eades, P.: Planarity for clustered graphs. In: Spirakis, P.G. (ed.) ESA 1995. LNCS, vol. 979, pp. 213–226. Springer, Heidelberg (1995)
8. Goodrich, M.T., Lueker, G.S., Sun, J.Z.: C-planarity of extrovert clustered graphs. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 211–222. Springer, Heidelberg (2006)
9. Gutwenger, C., Jünger, M., Leipert, S., Mutzel, P., Percan, M., Weiskircher, R.: Advances in c-planarity testing of clustered graphs. In: Goodrich, M.T., Kobourov, S.G. (eds.) GD 2002. LNCS, vol. 2528, pp. 220–235. Springer, Heidelberg (2002)
10. 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, pp. 261–272. Springer, Heidelberg (2003)
11. Hopcroft, J., Tarjan, R.E.: Efficient planarity testing. J. ACM 21(4), 549–568 (1974)
12. 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)

Appendix D

Monochromatic Triangles in Two-Colored Plane

COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA **29** (6) (2009) 699–718
DOI: 10.1007/s00493-009-2291-y

MONOCHROMATIC TRIANGLES IN TWO-COLORED PLANE

VÍT JELÍNEK, JAN KYNČL, RUDOLF STOLAŘ, TOMÁŠ VALLA

Received March 27, 2006

We prove that for any partition of the plane into a closed set C and an open set O and for any configuration T of three points, there is a translated and rotated copy of T contained in C or in O.

Apart from that, we consider partitions of the plane into two sets whose common boundary is a union of piecewise linear curves. We show that for any such partition and any configuration T which is a vertex set of a non-equilateral triangle there is a copy of T contained in the interior of one of the two partition classes. Furthermore, we characterize the "polygonal" partitions that avoid copies of a given equilateral triple.

These results support a conjecture of Erdős, Graham, Montgomery, Rothschild, Spencer and Straus, which states that every two-coloring of the plane contains a monochromatic copy of any nonequilateral triple of points; on the other hand, we disprove a stronger conjecture by the same authors, by providing non-trivial examples of two-colorings that avoid a given equilateral triple.

1. Introduction

Euclidean Ramsey theory addresses the problems of the following kind: assume that a finite configuration X of points is given; for what values of c and d is it true that every coloring of the d-dimensional Euclidean space by c colors contains a monochromatic congruent copy of X? The first systematic treatise on this theory appears in 1973 in a series of papers [2–4] by Erdős, Graham, Montgomery, Rothschild, Spencer and Straus. Since then, many strong results have been obtained in this field, often related to high-

Mathematics Subject Classification (2000): 05D10, 52C10 dimensional configurations (see, e.g., [5,7–9] or the survey [6]); however, there are basic "low-dimensional" problems that remain open.

In this paper, we study the configurations of three points in the Euclidean plane colored by two colors. We use the term *triangle* to refer to any set of three points, including collinear triples of points, which we call *degenerate* triangles. An (a,b,c)-triangle is a triangle whose edges, in anti-clockwise order, have respective lengths a, b and c. A $(1,1,1)$-triangle is also called a *unit triangle.*

We say that a set of points $X \subseteq \mathbb{R}^2$ is a *copy* of a set of points $Y \subseteq \mathbb{R}^2$, if X can be obtained from Y by translations and rotations in the plane. A *coloring* is a partition of $\mathbb{R}^2$ into two sets $\mathfrak{B}$ and $\mathfrak{W}$. The elements of $\mathfrak{B}$ and $\mathfrak{W}$ are called *black points* and *white points*, respectively. The term *boundary of* χ, denoted by $\Delta(\chi)$, refers to the common boundary of the sets $\mathfrak{B}$ and $\mathfrak{W}$. Given a coloring $\chi = (\mathfrak{B}, \mathfrak{W})$, we say that a set X is *monochromatic*, if $X \subseteq \mathfrak{B}$ or $X \subseteq \mathfrak{W}$.

We say that a coloring χ *contains* a triangle T, if there exists a monochromatic set T' which is a copy of T; otherwise, we say that χ *avoids* T.

A coloring that avoids the unit triangle is easy to obtain: consider a coloring χ^* that partitions the plane into alternating half-open strips of width $\frac{\sqrt{3}}{2}$; formally, a point (x,y) is black if and only if $n\sqrt{3} < y \leq (n+\frac{1}{2})\sqrt{3}$ for some integer n. It can be easily checked that χ^* avoids the unit triangle. We can even change the color of some of the points on the boundary of χ^* without creating any monochromatic unit triangle. Erdős et al. [4, Conjecture 1] conjectured that this is essentially the only example of colorings avoiding a given triangle:

Conjecture 1.1 ([4])**.** For every triangle T and every coloring χ, if χ avoids T, then T is an equilateral (l,l,l)-triangle and χ is equal to an l-times scaled copy of the coloring χ^* defined above, up to possible modifications of the colors of the points on the boundary of the strips.

In Section 3, we disprove this conjecture, and define a more general class of colorings that avoid the unit triangle. On the other hand, the following conjecture by Erdős et al. [4, Conjecture 3] remains open:

Conjecture 1.2 ([4])**.** Every coloring contains every nonequilateral triangle.

In the past, it has been shown that Conjecture 1.2 holds for special types of triangles T (see, e.g., [4,6,10]). Our approach is different: we prove that the conjecture is valid for a restricted class of colorings χ and arbitrary T. In Section 2, we show that every coloring that partitions $\mathbb{R}^2$ into a

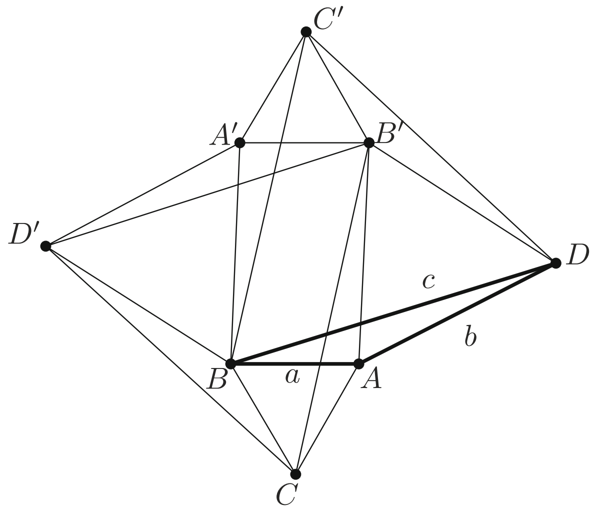


Fig. 1. The illustration of the proof of Lemma 1.3

closed set and an open set contains every triangle T. Then, in Section 3, we consider *polygonal* colorings, whose boundary is a union of piecewise linear curves (see page 704 for the precise definition). We show that Conjecture 1.2 holds for polygonal colorings, but there are polygonal counterexamples to the stronger Conjecture 1.1. In fact, we are able to characterize all these polygonal counterexamples.

The following lemma from [4] offers a useful insight into the topic of monochromatic triangles in two-colored plane:

Lemma 1.3 ([4]). *Let χ be a coloring of the plane. The following holds:*

(i) If χ contains an (a,a,a)-triangle for some $a > 0$, then χ contains an (a,b,c)-triangle for every $b,c > 0$ such that a,b,c satisfy the (possibly degenerate) triangle inequality.

(ii) If χ contains an (a,b,c)-triangle, then χ contains an (x,x,x)-triangle for some $x \in \{a,b,c\}$.

(iii) χ contains every triangle if and only if χ contains every equilateral triangle.

(iv) χ contains every non-equilateral triangle if and only if there is an $a_0 > 0$ such that χ contains the equilateral (a,a,a)-triangle for all values of $a > 0$ different from a_0.

(v) χ contains an (a,b,c)-triangle if and only if χ contains a (b,a,c)-triangle.

Proof. The essence of the proof is the configuration in Fig. 1. The configuration consists of two (a,a,a)-triangles ABC and $A'B'C'$, two (b,b,b)-triangles ADB' and $A'D'B$ and two (c,c,c)-triangles BDC' and $B'D'C$. To prove part (i), assume for a given χ, that there is a monochromatic (a,a,a)-triangle ABC, and extend this triangle into the eight-point configuration of

Fig. 1. It is easy to check that any coloring of the remaining five points of the configuration yields a monochromatic (a,b,c)-triangle. Part (ii) of the lemma is proved by an analogous argument, using the same configuration of points. Parts (iii), (iv) and (v) of the lemma are direct consequences of the first two parts. ∎

2. Colorings by closed and open sets

The aim of this section is to prove the following result:

Theorem 2.1. *If $\chi = (\mathfrak{B}, \mathfrak{W})$ is a coloring such that $\mathfrak{B}$ is closed and $\mathfrak{W}$ is open, then χ contains every triangle T.*

By Lemma 1.3, it suffices to prove Theorem 2.1 for the case when T is an arbitrary equilateral triangle. Moreover, since scaling does not affect the topological properties of $\mathfrak{B}$ and $\mathfrak{W}$, we may assume that T is the unit triangle. Before stating the proof, we introduce a definition and prove an auxiliary result.

Definition 2.2. Let $\varepsilon > 0$. An (a,b,c)-triangle whose edge-lengths satisfy $1-\varepsilon \leq a,b,c \leq 1+\varepsilon$ is called *an ε-almost unit triangle.*

Suppose that an orthogonal coordinate system is given in the plane. For $a > 0$, let $Q(a)$ be the closed square with vertices (a,a), $(-a,a)$, $(-a,-a)$, and $(a,-a)$.

Proposition 2.3. *Let $Q(2) = \mathfrak{B} \cup \mathfrak{W}$ be a partition of the square $Q(2)$ into two disjoint sets such that there is no monochromatic unit triangle in $Q(2)$. Then for every $\varepsilon > 0$ both $\mathfrak{B}$ and $\mathfrak{W}$ contain an ε-almost unit triangle.*

Proof. Assume that we are given a positive ε and a partition $\mathfrak{B} \cup \mathfrak{W} = Q(2)$ such that $Q(2)$ does not contain any monochromatic unit triangle. For contradiction, assume that one of the classes, wlog the class $\mathfrak{B}$, does not contain any ε-almost unit triangle.

There is a white point W and a black point B in $Q(1)$ such that $|W-B| < \varepsilon$ (otherwise the whole square $Q(1)$ would be monochromatic). Let $\mathcal{C}$ be the unit circle centered at W. For every $\alpha \in \mathbb{R}$, let $K(\alpha)$ denote the point of $\mathcal{C}$ with coordinates $(x_W + \cos(\alpha), y_W + \sin(\alpha))$, where (x_W, y_W) are the coordinates of W.

Note that the distance between B and any point on $\mathcal{C}$ is in the interval $(1-\varepsilon, 1+\varepsilon)$; thus, for every α, the points $K(\alpha)$ and $K(\alpha + \frac{\pi}{3})$ must have

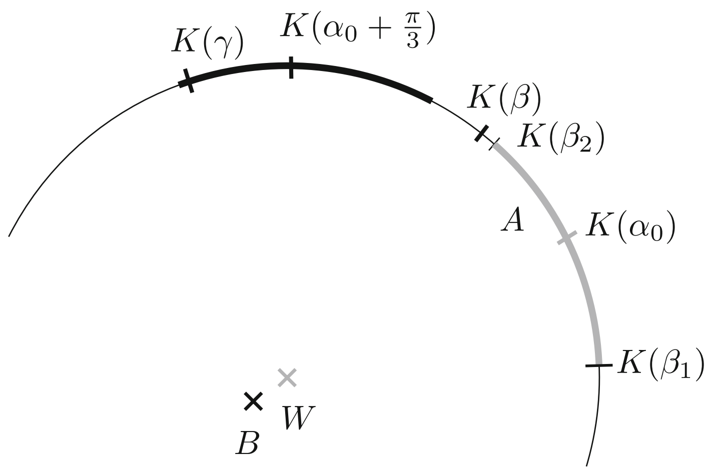


Fig. 2. Illustration of the proof of Proposition 2.3

different colors, otherwise they would form a monochromatic white unit triangle with W or a monochromatic black ε-almost unit triangle with B.

Let $K(\alpha_0)$ be a white point, then $K(\alpha_0+\frac{\pi}{3})$ is black (see Fig. 2). Note that for every $\alpha\in(\alpha_0-\varepsilon,\alpha_0+\varepsilon)$ the distance between $K(\alpha)$ and $K(\alpha_0+\frac{\pi}{3})$ is in the interval $(1-\varepsilon,1+\varepsilon)$, so the whole arc $\{K(\alpha);\alpha\in(\alpha_0-\varepsilon,\alpha_0+\varepsilon)\}$ is white. Let $A=\{K(\alpha);\alpha\in(\beta_1,\beta_2)\}$ be the maximal open white arc of $\mathfrak{C}$ containing the point $K(\alpha_0)$. Then the whole arc $\{K(\alpha);\alpha\in(\beta_1+\frac{\pi}{3},\beta_2+\frac{\pi}{3})\}$ is black. By definition of A, there exists $\beta\in(\beta_2,\beta_2+\frac{\varepsilon}{2})$ such that $K(\beta)$ is black. There also exists $\gamma\in(\beta_2+\frac{\pi}{3}-\frac{\varepsilon}{2},\beta_2+\frac{\pi}{3})$ such that $K(\gamma)$ is black. But then $(\gamma-\beta)\in(\frac{\pi}{3}-\varepsilon,\frac{\pi}{3})$, so the distance between the black points $K(\beta)$ and $K(\gamma)$ is in the interval $(1-\varepsilon,1)$, and the three points $B,K(\beta),K(\gamma)$ form a black ε-almost unit triangle – a contradiction. ∎

We are now ready to prove the main result of this section.

Proof of Theorem 2.1. Let $\chi=(\mathfrak{B},\mathfrak{W})$ be a coloring, with $\mathfrak{B}$ closed. By the remark under Theorem 2.1, we only need to show that χ contains the unit triangle. Assume that this is not the case. Let $\mathfrak{B}_0=Q(2)\cap\mathfrak{B}$ and $\mathfrak{W}_0=Q(2)\cap\mathfrak{W}$. Neither $\mathfrak{B}_0$ nor $\mathfrak{W}_0$ contain the unit triangle, so by Proposition 2.3, both these sets contain ε-almost unit triangles for every $\varepsilon>0$. In particular, the set $\mathfrak{B}_0$ contains, for every $n\in\mathbb{N}$, a $\frac{1}{n}$-almost unit triangle $X_nY_nZ_n$.

Since $\mathfrak{B}_0$ is a compact set, the set $\mathfrak{B}_0^3=\mathfrak{B}_0\times\mathfrak{B}_0\times\mathfrak{B}_0$ is compact as well. The sequence $\{(X_n,Y_n,Z_n);n\in\mathbb{N}\}$ is an infinite sequence of points in $\mathfrak{B}_0^3$, so it has a convergent subsequence $\{(X_{n_k},Y_{n_k},Z_{n_k});k\in\mathbb{N}\}$. Let $(X,Y,Z)\in\mathfrak{B}_0^3$ be its limit. Then $X,Y,Z\in\mathfrak{B}$ are limits of the sequences $\{X_{n_k};k\in\mathbb{N}\}$, $\{Y_{n_k};k\in\mathbb{N}\}$, and $\{Z_{n_k};k\in\mathbb{N}\}$, respectively. The Euclidean distance is a continuous function of two variables, so $|X-Y|=\lim_{k\to\infty}|X_{n_k}-Y_{n_k}|=1$, similarly $|Y-Z|=|Z-X|=1$. Thus, $\{X,Y,Z\}$ is a black unit triangle in $Q(2)$, which is a contradiction. ∎

3. Polygonal colorings

Throughout this section, $\mathcal{C}(A)$ denotes the unit circle with center A, and $\mathcal{D}(A)$ denotes the closed unit disc with center A.

In this section, we consider *polygonal* colorings, defined as follows:

Definition 3.1. A coloring $\chi=(\mathfrak{B},\mathfrak{W})$ is said to be *polygonal*, if it satisfies the following conditions:

- The boundary of χ is a union of Jordan curves. Each of these curves is either closed or two-way unbounded. Furthermore, each of these curves is *piecewise linear*, i.e., it is a union of (possibly unbounded) linear segments. We will call these segments *boundary segments*. Two boundary segments may only intersect in their common endpoint. The endpoints of the boundary segments will be called *boundary vertices*. We assume that if exactly two boundary segments meet at a boundary vertex, then the two segments do not form a straight angle, because otherwise we could replace them with a single boundary segment.
- The arrangement of boundary segments is *locally finite*, i.e., every bounded region of the plane is intersected by only finitely many boundary segments (which implies that every bounded region contains only finitely many boundary vertices).
- The union of all the boundary segments partitions the plane into cells whose interiors are monochromatic. Each two cells that share a common boundary segment have different colors (it is not difficult to see that such a two-coloring of the cells exists for any collection of boundary curves satisfying the conditions above). We make no assumptions about the colors of the points on the boundary.

We say that a coloring χ' is a *twin* of a coloring χ if $\Delta(\chi)=\Delta(\chi')$ and the two colorings assign the same colors to the points outside their boundary.

The main aim of this section is to prove that each polygonal coloring contains each nonequilateral triangle, and to characterize the polygonal colorings that avoid an equilateral triangle. To achieve this, we need the following definition:

Definition 3.2. A coloring $\chi=(\mathfrak{B},\mathfrak{W})$ is called *zebra-like* if $\Delta(\chi)$ is a disjoint union of infinitely many Jordan curves $\mathcal{L}_i; i\in\mathbb{Z}$ with the following properties (see Fig. 3):

(a) There is a unit vector $\vec{x}$ such that for every $i \in \mathbb{Z}$, $\mathcal{L}_i + \vec{x} = \mathcal{L}_i$. In other words, each $\mathcal{L}_i$ is invariant upon a translation of length 1.

(b) There is a unit vector $\vec{y}$ which forms an angle of size $\frac{\pi}{3}$ with $\vec{x}$, such that for every $i \in \mathbb{Z}$, the curve $\mathcal{L}_{i+1}$ is a translated copy of $\mathcal{L}_i$, and the translation is given by the vector $\vec{y}$. In other words, if $X \in \mathcal{L}_i$, then $X + \vec{x} \in \mathcal{L}_i$ and $X + \vec{y} \in \mathcal{L}_{i+1}$. Note that the points $X, X+\vec{x}, X+\vec{y}$ form a unit triangle.

(c) For every $i \in \mathbb{Z}$, the interior of the region delimited by $\mathcal{L}_i \cup \mathcal{L}_{i+1}$ is colored with a different color than the interior of the region delimited by $\mathcal{L}_{i-1} \cup \mathcal{L}_i$.

(d) Let $A \in \mathcal{L}_i$ be an arbitrary point of $\Delta(\chi)$. Let $B_1 = A - \vec{x} + \vec{y}$, $B_2 = A + \vec{y}$, $C_1 = A - \vec{y}$ and $C_2 = A + \vec{x} - \vec{y}$ (note that $B_1, B_2 \in \mathcal{L}_{i+1}$ and $C_1, C_2 \in \mathcal{L}_{i-1}$ by the previous conditions, and that all these four points are at a unit distance from A). Under these assumptions, the portion of $\mathcal{L}_{i+1}$ between B_1 and B_2 and the portion of $\mathcal{L}_{i-1}$ between C_1 and C_2 are both contained in the closed unit disk $\mathcal{D}(A)$, and no other point of $\mathcal{L}_{i+1} \cup \mathcal{L}_{i-1}$ belongs to $\mathcal{D}(A)$.
This last condition can also be stated in the following equivalent form: for two points A and B, let θ_{AB} denote the size of the acute angle formed by the segment AB and the vector $\vec{x}$. For every $i \in \mathbb{Z}$ and every two points $A \in \mathcal{L}_i$ and $B \in \mathcal{L}_{i+1}$, the inequality $\|AB\| > 1$ holds if and only if $\theta_{AB} < \frac{\pi}{3}$.

We stress that a zebra-like coloring is not necessarily polygonal.

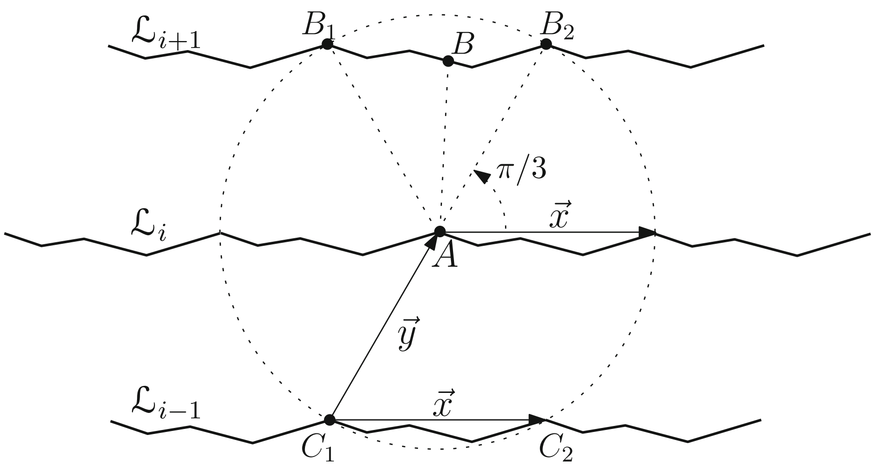


Fig. 3. Boundary of a zebra-like coloring

3.1. The main result

The following theorem is the main result of this section:

Theorem 3.3. *For a polygonal coloring χ, these conditions are equivalent:*

(C1) The coloring χ is zebra-like.
(C2) The coloring χ has a twin that avoids the unit triangle.
(C3) For every monochromatic unit triangle ABC, at least one of the three points A, B and C belongs to the boundary of χ.

Clearly, condition (C2) of Theorem 3.3 implies condition (C3), so we only need to prove that (C1) implies (C2) and that (C3) implies (C1).

The proof is organized as follows: we first prove that (C3)⇒(C1). This part of the proof proceeds in several steps: first of all, we use condition (C3) to describe the set $\mathit{\Delta}(\chi)\cap\mathcal{C}(A)$, where A is a boundary point. Then we apply a continuity argument to extend this information into a global description of χ.

Next, in Theorem 3.19, we prove that every (not necessarily polygonal) zebra-like coloring has a twin that avoids the unit triangle, which shows that (C1)⇒(C2), completing the proof of Theorem 3.3.

In the last part of this section, we show that Theorem 3.3 implies that every polygonal coloring contains a monochromatic copy T of a given non-equilateral triangle, with the vertices of T avoiding the boundary.

3.2. Proof of the main result

We begin with an auxiliary lemma:

Lemma 3.4. *Let q_1, q_2, q_3 be (not necessarily distinct) lines in the plane, not all three parallel. Exactly one of the following possibilities holds:*

1. *The lines q_1, q_2, q_3 intersect at a common point and every two of them form an angle $\frac{\pi}{3}$.*
2. *There exist only finitely many unit triangles ABC such that $A \in q_1$, $B \in q_2$ and $C \in q_3$.*

Proof. It can be easily checked that the two conditions cannot hold simultaneously: in fact, if the three lines satisfy the first condition, then for every point $A \in q_1$ whose distance from the other two lines is at most 1 there are points $B \in q_2$ and $C \in q_3$ such that ABC is a unit triangle. We now show that at least one of the two conditions holds.

We may assume that neither q_1 nor q_2 is parallel to q_3. Consider a Cartesian coordinate system whose y-axis is q_3. There are real numbers a_1, a_2, b_1, b_2 such that for $i \in \{1,2\}$ we have $q_i = \{(x,y) \in \mathbb{R}^2; y = a_i x + b_i\}$. Let ABC be a unit triangle with $A = (x_1, y_1) \in q_1$, $B = (x_2, y_2) \in q_2$ and $C \in q_3$, and assume that A, B, C are in the counter-clockwise order (the other case is symmetric). Then $C = (\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}) + \frac{\sqrt{3}}{2}(y_1 - y_2, x_2 - x_1)$. Since C lies on q_3, we have

$$\frac{x_1 + x_2}{2} + \frac{\sqrt{3}(y_1 - y_2)}{2} = 0, \tag{1}$$

and since A and B are at distance 1, we get

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 = 1. \tag{2}$$

By combining (1) and (2), eliminating y_1, y_2, and simplifying, we obtain

$$x_1^2 + x_2^2 - x_1x_2 = \frac{3}{4}. \tag{3}$$

Substituting $y_1 = a_1x_1 + b_1$ and $y_2 = a_2x_2 + b_2$ into (1) gives

$$\frac{1 + \sqrt{3}a_1}{2}x_1 + \frac{1 - \sqrt{3}a_2}{2}x_2 + \frac{\sqrt{3}}{2}(b_1 - b_2) = 0. \tag{4}$$

If both $\frac{1+\sqrt{3}a_1}{2}$ and $\frac{1-\sqrt{3}a_2}{2}$ are equal to zero, we get $a_1 = -\frac{1}{\sqrt{3}}$, $a_2 = \frac{1}{\sqrt{3}}$ and $b_1 = b_2$, and the first case of the statement holds.

In the other case, suppose (wlog) that $\frac{1+\sqrt{3}a_1}{2} \neq 0$. From (4) we obtain that $x_1 = cx_2 + d$ for some reals c, d. Substituting this into (3) yields a quadratic equation for the variable x_2 whose leading coefficient is equal to $c^2 - c + 1 = (c - \frac{1}{2})^2 + \frac{3}{4} > 0$, so there are at most two possible values for x_2, thus at most two possible locations of B and at most four possible unit triangles ABC. ∎

Throughout the rest of this section, we assume that χ is a fixed polygonal coloring satisfying condition (C3) of Theorem 3.3, let $\Delta = \Delta(\chi)$ be its boundary. A boundary segment is as a common edge of two (possibly unbounded) polygonal regions, one of which is white and the other black. We choose an orientation of the boundary segments in the following way: a boundary segment with endpoints A and B is directed from A to B if the white region adjacent to this segment is on the left hand side from the point of view of an observer walking from A to B.

Definition 3.5. A boundary point A is called *feasible*, if A is not a boundary vertex, and the unit circle $\mathcal{C}(A)$ does not contain any boundary vertex. An *infeasible* point is a point on Δ that is not feasible.

We may easily see that every bounded subset of the plane contains only finitely many infeasible points.

The first step in the proof of the main result is the description of the set of all boundary points at unit distance from a given feasible point A.

Let A be a fixed feasible point, let s be the boundary segment containing A. The set $\Delta \cap \mathcal{C}(A)$ is finite, by the definition of polygonal coloring; on the other hand, this set is nonempty, otherwise we could find two points B, C of $\mathcal{C}(A)$ such that ABC is a unit triangle, with B and C in the interior of the same color class. By shifting the triangle ABC slightly in a suitable direction, we would obtain a monochromatic unit triangle avoiding the boundary, which is forbidden by condition (C3).

In the following arguments, we will use a Cartesian coordinate system whose origin is the point A, and whose x-axis is parallel to s and has the same orientation. We shall assume that the x-axis and the segment s is directed left-to-right and the y-axis is directed bottom-to-top. Assuming this coordinate system, we let $P(\alpha, A)$ denote the point of $\mathcal{C}(A)$ with coordinates $(\cos(\alpha), \sin(\alpha))$. If no ambiguity arises, we write $P(\alpha)$ instead of $P(\alpha, A)$.

Lemma 3.6. *Let $B = P(\alpha)$ be an arbitrary element of $\Delta \cap \mathcal{C}(A)$, let t be the boundary segment containing B (t is determined uniquely, because A is a feasible point). Then the segments s and t are parallel.*

Proof. Assume that s and t are not parallel, let $\sigma \in (0, \pi)$ be the angular slope of t with respect to the coordinate system established above, i.e., σ is the angle formed by the lines containing s and t.

First of all, note that point $C = P(\alpha + \frac{\pi}{3})$ lies on Δ; otherwise, a sufficiently small translation of the unit triangle ABC in a suitable direction would yield a counterexample to condition (C3) (here we use the assumption that s and t are not parallel). Let u be the boundary segment containing C, and let τ be the angular slope of u.

Secondly, we deduce that $\{\sigma, \tau\} = \{\frac{\pi}{3}, \frac{2\pi}{3}\}$, and the three lines containing s, t and u all meet at one point. If this were not the case, then by Lemma 3.4 there would be only finitely many unit triangles with vertices belonging to the three segments s, t and u. Thus, we could find a unit triangle $A'B'C'$ with $A' \in s$, $B' \in t$ and $C' \notin \Delta$, which is impossible, by the argument from the previous paragraph. By repeating this argument with $\{\alpha + \frac{i\pi}{3};\ i = 1, \dots, 5\}$ in place of α, we obtain the following conclusions:

- The six points $\{P(\alpha + \frac{i\pi}{3});\ i = 1, \dots, 6\}$ all belong to Δ.
- The lines passing through the boundary segments containing these six points all meet at one point.

- The boundary segments containing $P(\alpha)$, $P(\alpha+\frac{2\pi}{3})$ and $P(\alpha+\frac{4\pi}{3})$ all have the same slope.

This is a contradiction, because three parallel segments intersecting a circle in three distinct points cannot belong to a single line, and two parallel lines do not intersect. ∎

Lemma 3.7. $P(\frac{\pi}{2})\notin\Delta$, $P(-\frac{\pi}{2})\notin\Delta$.

Proof. For contradiction, assume that $B=P(\frac{\pi}{2})\in\Delta$ (the case of $P(-\frac{\pi}{2})$ is symmetric), let t denote the boundary segment containing B. Let $C=P(\frac{\pi}{6})$. We distinguish the following cases:

- Segment t has the same orientation as segment s. In this case, by applying a rotation around the center C and then, if $C\in\Delta$, a suitable translation, we transform the triple ABC into a monochromatic triple with vertices avoiding the boundary, contradicting (C3).
- Segments s and t have opposite orientations (i.e., t is oriented right-to-left, which means that there is a white region touching t from below); furthermore, either $C\in\Delta$ or C is in the interior of the white color. In such case, we may rotate the configuration ABC around the center of the segment AB to obtain a unit triangle in the interior of the white color.
- Segments s and t have opposite orientations and the point C is in the interior of the black color (see Fig. 4). Let θ be the maximal angle with the properties that for every $\alpha\in(\frac{\pi}{2},\frac{\pi}{2}+\theta)$ the point $P(\alpha)$ is in the interior of the white color and for every $\alpha\in(\frac{\pi}{6},\frac{\pi}{6}+\theta)$ the point $P(\alpha)$ is in the interior of the black color. The value of θ is well defined, and by the previous assumptions, $0<\theta<\frac{\pi}{3}$. Let $B'=P(\frac{\pi}{2}+\theta)$ and $C'=P(\frac{\pi}{6}+\theta)$. By the maximality of θ, at least one of the two points lies on the boundary, and the boundary segment passing through this point is directed left-to-right. As in the first case of this proof, we may rotate and translate the configuration $AB'C'$ to get a contradiction.

The lemma follows. ∎

The previous two lemmas imply that if A is a feasible point, then no boundary segment is tangent to $\mathcal{C}(A)$.

Lemma 3.8. *Let $B=P(\alpha)\in\Delta$ be a point on the boundary, let t be the boundary segment containing B. If $\alpha\in(\frac{\pi}{6},\frac{5\pi}{6})\cup(\frac{7\pi}{6},\frac{11\pi}{6})$, then s and t have opposite orientation, while if $|\alpha|<\frac{\pi}{6}$ or $|\alpha-\pi|<\frac{\pi}{6}$, then s and t have the same orientation.*

Proof. We first consider the case $\alpha\in(\frac{\pi}{6},\frac{5\pi}{6})\cup(\frac{7\pi}{6},\frac{11\pi}{6})$. The proof is analogous to the proof of the first part of Lemma 3.7: if t had the same orientation

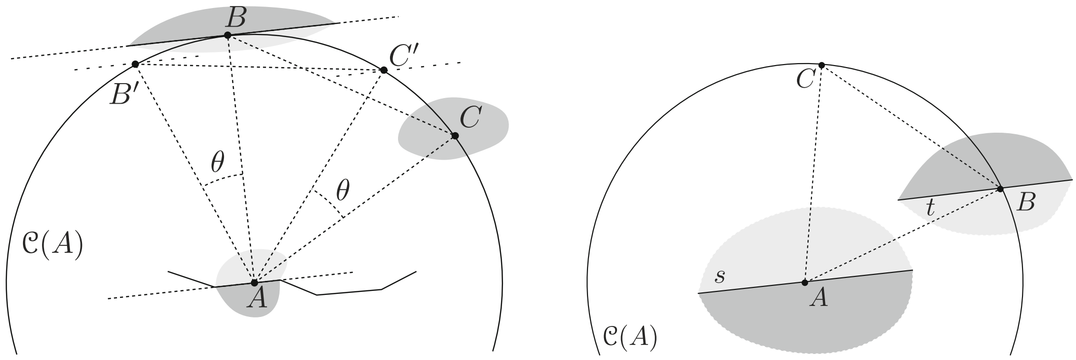


Fig. 4. Illustration of the proof of Lemma 3.7 (left) and Lemma 3.9 (right)

as s, we could take $C=P(\frac{\pi}{3}+\alpha)$ and then by rotating and translating the unit triangle ABC we would get a contradiction. Note that the condition $\alpha\in(\frac{\pi}{6},\frac{5\pi}{6})\cup(\frac{7\pi}{6},\frac{11\pi}{6})$ guarantees that C is either the leftmost or the rightmost point of the triangle ABC, so whenever we start rotating the triangle ABC around C, the two points A,B move into the interior of the same color.

The case $|\alpha|<\frac{\pi}{6}$ or $|\alpha-\pi|<\frac{\pi}{6}$ can be proven analogously. ∎

Lemma 3.9. *$P(\alpha)\in\varDelta$ if and only if $P(\alpha+\frac{\pi}{3})\in\varDelta$.*

Proof. We only prove one implication, the other case is symmetric. Assume that for some α we have $P(\alpha)\in\varDelta$ and $P(\frac{\pi}{3}+\alpha)\notin\varDelta$. Let $B=P(\alpha)$, $C=P(\frac{\pi}{3}+\alpha)$, and let t be the boundary segment containing B. We consider the following cases:

- If s and t have opposite orientation, we rotate ABC around the center of AB to obtain a monochromatic unit triangle in the interior of one color (see Fig. 4). We use the fact that $\alpha\neq\frac{\pi}{2}$, which follows from Lemma 3.7.
- If s and t have the same orientation, a small translation in a suitable direction transforms ABC into a monochromatic unit triangle.

In both cases we get a contradiction. ∎

Lemma 3.10. *For every θ there is exactly one value of $\alpha\in[\theta,\theta+\frac{\pi}{3})$ such that $P(\alpha)\in\varDelta$.*

Proof. By Lemma 3.9, if the statement holds for some value of θ, it holds for all other values of θ as well. Thus, it is enough to prove the lemma for $\theta=\frac{\pi}{2}$.

Clearly, there is at least one $\alpha\in[\frac{\pi}{2},\frac{5\pi}{6})$ such that $P(\alpha)\in\varDelta$; otherwise, the set $\mathcal{C}(A)\cap\varDelta$ would be empty, which is impossible.

Assume that there are α and α' such that $\frac{\pi}{2}\leq\alpha<\alpha'<\frac{5\pi}{6}$ with $P(\alpha)\in\varDelta$ and $P(\alpha')\in\varDelta$. Fix α and α' as small as possible. Let t and t' be the boundary segments containing $P(\alpha)$ and $P(\alpha')$. The circle $\mathcal{C}(A)$ consists of alternating black and white arcs and one of these arcs has $P(\alpha)$ and $P(\alpha')$ for endpoints. It follows that one of the segments t, t' has the same orientation as the segment s, contradicting Lemma 3.8. ∎

Before we proceed with the proof of the main result, we summarize the lemmas proved so far (and introduce some related notation) in the following claim (see Fig. 5):

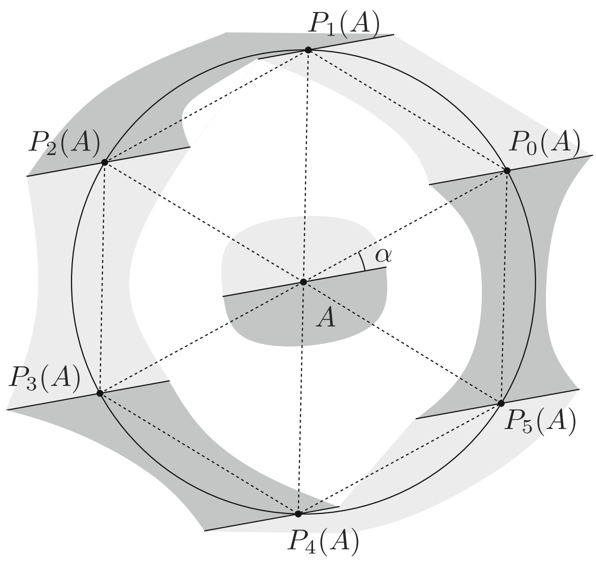


Fig. 5. Illustration of Claim 3.11

Claim 3.11. *Let $A \in \Delta$ be an arbitrary feasible point. The circle $\mathcal{C}(A)$ intersects the boundary Δ at exactly six points, which form the vertex set of a regular hexagon. These six points will be denoted by $P_0(A), \ldots, P_5(A)$, where $P_i(A) = P(\alpha + \frac{i\pi}{3}, A)$ with $\alpha \in (-\frac{\pi}{6}, \frac{\pi}{6})$ (this determines $P_i(A)$ uniquely). The boundary segments containing the six points $P_i(A)$ are all parallel to the boundary segment s containing the point A. The boundary segments containing the points $P_0(A)$ and $P_3(A)$ have the same orientation as s, whereas the boundary segments containing $P_1(A)$, $P_2(A)$, $P_4(A)$ and $P_5(A)$ have opposite orientation.*

Now we use Claim 3.11 to obtain global information about the boundary.

Lemma 3.12. *Let u_1 and u_2 be two boundary segments with a common endpoint X. The size of the convex angle formed by the two segments is greater than $\frac{2\pi}{3}$.*

Proof. For contradiction, assume that for some u_1, u_2 and X, the statement of the lemma does not hold (see Fig. 6). We may assume that the convex angle determined by u_1 and u_2 does not contain any other boundary segment with endpoint X. Furthermore, we may assume that the segment u_1 is directed from X to the other endpoint.

For $0 < t < |u_1|$, let $A(t) \in u_1$ denote the point with $|A(t) - X| = t$ and let $A'(t) = P_4(A(t))$. There is an $\varepsilon > 0$ such that for all $0 < t < \varepsilon$ the points $A(t)$ are feasible, the points $A'(t)$ are feasible as well and lie on a common boundary segment. By our assumption, the convex angles between the ray $A(t)A'(t)$ and the segments u_1, u_2 directed from X are both greater than $\frac{\pi}{2}$. It follows that if t is sufficiently small, the tangent line to the circle $\mathcal{C}(A'(t))$ at $A(t)$ intersects both segments u_1, u_2 and so does the circle $\mathcal{C}(A'(t))$, contradicting Claim 3.11. ∎

Lemma 3.12 shows that no three boundary segments share a common endpoint. Thus, each connected component of the boundary is a piecewise linear curve (either open or closed). We will call these curves *boundary components*.

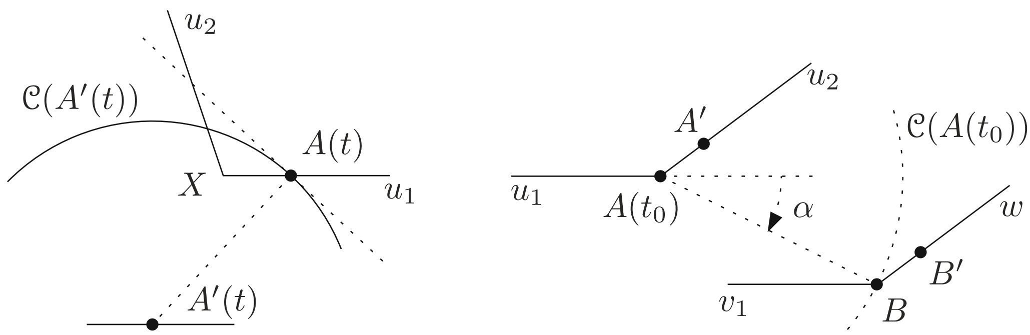


Fig. 6. Illustration of the proof of Lemma 3.12 (left) and Lemma 3.14 (right)

Definition 3.13. Let $A \in \Delta$. For $t \in \mathbb{R}$, let $A(t)$ denote the point of the same boundary component as A, such that the directed length of the part of the boundary starting at A and ending at $A(t)$ is equal to t. $A(t)$ is clearly a continuous function of t. If $A(t)$ is a feasible point, we let $p_i(t) = P_i(A(t))$, for $i = 0, \ldots, 5$.

It is easy to see that the functions p_i are continuous on a sufficiently small neighborhood of every value of t for which $A(t)$ is a feasible point. Our next aim is to show that these functions can be extended into continuous functions by suitably defining the values of $p_i(t)$ when $A(t)$ is not feasible. It is not obvious that the functions p_i can be extended in this way: the definition of $P_i(A(t))$ uses the Cartesian system whose x-axis is parallel with the boundary segment containing $A(t)$. Hence, if A_1 and A_2 are two feasible points belonging to two distinct boundary segments of the same boundary component, it might not be immediately clear that $P_i(A_1)$ belongs to the same boundary component as $P_i(A_2)$. The next lemma shows that these technical difficulties can be overcome.

Lemma 3.14. *Let $A(t_0)$ be an infeasible point. For every $i=0,\dots,5$, there is a point $P_i \in \Delta$ such that $\lim_{t\to t_0-} p_i(t) = P_i = \lim_{t\to t_0+} p_i(t)$.*

This means that if we define $p_i(t_0)=P_i$, then p_i is continuous at t_0.

Proof. It is sufficient to prove the lemma for $i=0$, because $p_i(t)$ is a continuous function of $A(t)$ and $p_0(t)$. Since every boundary segment contains only finitely many infeasible points, we may choose an $\varepsilon>0$, such that for every t from the open interval $(t_0-\varepsilon, t_0)$ the points $A(t)$ are feasible and they all belong to a single boundary segment u_1, and similarly, for every $t' \in (t_0, t_0+\varepsilon)$ the points $A(t')$ are feasible, and belong to a single boundary segment u_2. If the segments u_1 and u_2 are distinct, then $A(t_0)$ is their common endpoint. Note that for $t \in (t_0-\varepsilon, t_0)$, the points $p_0(t)$ belong to a single boundary segment v_1, otherwise some of the $A(t)$ would be infeasible. By Claim 3.11, the segment v_1 is parallel and consistently oriented with u_1. Similarly, for $t' \in (t_0, t_0+\varepsilon)$ the points $p_0(t')$ belong to a single boundary segment v_2, parallel and consistently oriented with u_2. We do not know yet whether v_1 and v_2 appear consecutively on the same component of the boundary.

Let $B=\lim_{t\to t_0-} p_0(t)$. See Fig. 6.

For $t \in (t_0-\varepsilon, t_0)$, fix $\alpha \in (-\frac{\pi}{6}, \frac{\pi}{6})$ such that $p_0(t) = P(\alpha, A(t))$, i.e., α is the (signed) measure of the angle between the segment u_1 and the segment $A(t)p_0(t)$. Note that α does not depend on the choice of t. The circle $\mathcal{C}(A(t_0))$ intersects Δ at B. Let w be the boundary segment starting at B and directed away from B. By Lemma 3.12, the convex angles determined by v_1 and w and by u_1 and u_2 have size at least $\frac{2\pi}{3}$, which implies that the convex angle α' between u_2 and $BA(t_0)$ is acute and the convex angle between w and $BA(t_0)$ is obtuse. Thus, for $t' \in (t_0, t_0+\varepsilon)$ the circle $\mathcal{C}(A')$ (where $A'=A(t')$) intersects the segment w at a point $B'=p_i(t')$. From Claim 3.11 it follows that w is parallel to u_2. Also, the segment $A'B'$ is parallel to the segment $A(t_0)B$, which is in turn parallel to any of the segments $A(t)p_0(t)$, for $t\in(t_0-\varepsilon, t_0)$.

To finish the proof, we need to show that $B' = p_0(t')$ (as opposed to $B'=p_i(t')$ for some $i\neq 0$), i.e., we need to prove that the angle α' determined by the segment u_2 and the segment $A'B'$ falls into the range $(-\frac{\pi}{6}, \frac{\pi}{6})$. We have observed that $\alpha' \in (-\frac{\pi}{2}, \frac{\pi}{2})$. This leaves us with the following three possibilities: either $B' = p_5(t')$, or $B' = p_1(t')$, or $B' = p_0(t')$. However, the former two possibilities are ruled out by the fact that the segment w is oriented consistently with the segment u_2. This concludes the proof. ∎

Lemma 3.15. *Let $i \in \{0,\dots,5\}$, let $A \in \Delta$ be an arbitrary boundary point. All the unit segments of the form $A(t)p_i(t)$ have the same slope, independently of the choice of t.*

Proof. The slope of $A(t)p_i(t)$ (as a function of t) is constant in a neighborhood of every t for which $A(t)$ is feasible. Moreover, this slope is a continuous function of t, which follows from Lemma 3.14. Hence the function is constant on the whole range. ∎

Lemma 3.15 shows that every translation that maps a feasible point A to the point $P_i(A)$ also maps the boundary component containing A onto the boundary component containing $P_i(A)$ (which may be the same component). Composing such translations (or their inverses) we conclude that the translations that send $P_i(A)$ to $P_j(A)$ have the same component-preserving property.

For the proof of Lemma 3.17, we will need a slight extension of Claim 3.11 to infeasible points:

Claim 3.16. *Let $A \in \Delta$ be an arbitrary infeasible point.*

(i) At each of the six points $P_0(A), P_1(A), \dots, P_5(A)$ the circle $\mathcal{C}(A)$ properly crosses the corresponding boundary component, i.e., in a sufficiently small neighborhood of such point, the circle $\mathcal{C}(A)$ separates the boundary component into two portions, one lying inside $\mathcal{C}(A)$ and the other one lying outside $\mathcal{C}(A)$.

(ii) There are no more proper crossings of $\mathcal{C}(A)$ with boundary components. (However, $\mathcal{C}(A)$ may touch the boundary at some other points.)

(iii) The boundary components containing the points $P_0(A)$ and $P_3(A)$ have the same orientation as the component containing A, while the boundary components containing $P_1(A)$, $P_2(A)$, $P_4(A)$ and $P_5(A)$ have opposite orientation.

Proof. The first two statements follow from the fact that $\mathcal{C}(A)$ has the same number of proper crossings with the boundary as the circle $\mathcal{C}(A(t))$, where $A(t)$ is a feasible point sufficiently close to A. The third statement follows from Claim 3.11 applied to point $A(t)$. ∎

Lemma 3.17. *Let $A \in \Delta$ be an arbitrary boundary point. For simplicity, write P_i, $\mathcal{C}$ and $\mathcal{D}$ instead of $P_i(A)$, $\mathcal{C}(A)$ and $\mathcal{D}(A)$, respectively. The point*

P_1 belongs to the same boundary component as P_2, the point P_0 belongs to the same boundary component as A and P_3, and the point P_4 belongs to the same boundary component as P_5. The four portions of the boundary that connect P_1 with P_2, P_0 with A, A with P_3, and P_4 with P_5 are all translated copies of a single piecewise linear curve. These four portions of the boundary are all contained in $\mathcal{D}$.

Proof. We show that the boundary component that enters inside $\mathcal{D}$ at P_1 leaves $\mathcal{D}$ at P_2. The rest of the statement follows from Lemma 3.15.

Let $\mathcal{L}$ be the boundary component that contains P_1. Let us follow $\mathcal{L}$ from P_1 in the direction of its orientation, i.e., into the interior of the unit disc $\mathcal{D}$, and let X be the first point where $\mathcal{L}$ leaves $\mathcal{C}$. We observe the following:

- X is neither P_3 nor P_5, because in these points, the boundary is oriented into the interior of the disc $\mathcal{D}$.
- X is not the point P_0: if $X = P_0$, then the translation $P_0 \mapsto A$ would map the section of the boundary between P_1 and P_0 onto a section directed from P_2 to A. Similarly, the translation $P_1 \mapsto A$ would map the section P_1P_0 onto a section directed from P_5 to A. This is impossible, because two different boundary sections of equal length cannot both end at A.
- X is not P_4: if X were equal to P_4, we would consider the boundary component that enters into the interior of $\mathcal{C}$ at P_3. Since this boundary component cannot intersect the boundary section between P_1 and P_4, it must leave the interior of $\mathcal{C}$ at the point P_2. However, this is symmetric to the previous case.
- Having excluded all other possibilities, we know that $X = P_2$.

Let U denote the section of $\mathcal{L}$ between P_1 and P_2. By definition, this section properly crosses $\mathcal{C}$ only at its endpoints. Applying a symmetric argument, we find that the boundary section from P_5 to P_4 (which is a translated copy of U) properly crosses $\mathcal{C}$ only at its endpoints. Translating U appropriately, we obtain the boundary sections connecting P_3 with A and A with P_0. ∎

From the previous lemmas, we readily obtain the following claim.

Claim 3.18. *Condition (C3) of Theorem 3.3 implies condition (C1).*

Proof. We check that χ satisfies the conditions of Definition 3.2. Let $\vec{x}$ denote the unit vector $\overrightarrow{AP_0}$ and let $\vec{y}$ be the unit vector $\overrightarrow{AP_1}$. As we know from Claim 3.11 and Lemma 3.15, the two vectors $\vec{x}$ and $\vec{y}$ form an angle $\frac{\pi}{3}$.

By Lemma 3.17, every component of Δ is a piecewise linear $\vec{x}$-periodic curve, and if $\mathcal{L}$ is a component of Δ, then any other component is a translate of $\mathcal{L}$ by an integral multiple of the vector $\vec{y}$. Let $\mathcal{L}_i = \mathcal{L}_0 + i\vec{y}$, $i \in \mathbb{Z}$, where

$\mathcal{L}_0$ is a boundary component chosen arbitrarily. We have $\Delta = \bigcup_{i\in\mathbb{Z}} \mathcal{L}_i$. Condition (d) of Definition 3.2 follows from Lemma 3.17. ∎

It remains to show that condition (C1) implies (C2). In fact, we prove a more general claim:

Theorem 3.19. *Each zebra-like coloring has a twin that avoids the unit triangle.*

Proof. Let χ be a zebra-like coloring, let $\mathcal{L}_i$, $\vec{x}$ and $\vec{y}$ be as in Definition 3.2. Let χ' be the twin coloring of χ such that the points of $\mathcal{L}_i$ are black in χ' if i is even and white if i is odd.

Observe that by the definition of the coloring, the color of a point P is equal to the color of $P+\vec{x}$ and different from the color of $P+\vec{y}$. Now assume that ABC is a monochromatic unit triangle, wlog the three points are black. By the previous observation, no edge of the triangle forms an angle of size $\frac{\pi}{3}$ (or $\frac{2\pi}{3}$) with the vector $\vec{x}$. It follows that exactly one of the three edges (wlog the edge AB) forms with $\vec{x}$ an angle whose size falls into the range $(\frac{\pi}{3}, \frac{2\pi}{3})$.

We claim that the three points A, B, C all belong to a single connected component of the black color: otherwise one of the two edges AC and BC would have to intersect (at least) two curves $\mathcal{L}_i$ and $\mathcal{L}_{i+1}$. By the definition of the coloring, the distance between the two points of intersection is greater than 1, contradicting the fact that ABC is a unit triangle.

We now show that $\|AB\| < 1$: let ℓ be the line containing the segment AB. Note that the line ℓ, as well as any other line not parallel with $\vec{x}$, intersects all the curves $\mathcal{L}_i$. Let $A'B'$ be the segment obtained as the convex hull of the intersection of ℓ with the closure of the black component containing A and B. By the definition of χ', $\|A'B'\| \leq 1$. Since the two points A' and B' belong to two adjacent boundary curves $\mathcal{L}_i$ and $\mathcal{L}_{i+1}$, they have different colors. Thus, the segment AB is a proper subset of $A'B'$, and $\|AB\| < 1$, which is a contradiction. ∎

This completes the proof of Theorem 3.3.

3.3. Nonequilateral triangles

The following result is a direct consequence of Theorem 3.3, by an easy modification of the proof of Lemma 1.3.

Theorem 3.20. *Let XYZ be a nonequilateral triangle, let χ be a polygonal coloring. There is a monochromatic copy $X'Y'Z'$ of the triangle XYZ, such that none of the points X', Y' and Z' belongs to the boundary of χ.*

Proof. Let a,b and c be the lengths of the three edges of XYZ, assume that $a \neq b$. By Theorem 3.3, no polygonal coloring can simultaneously avoid copies of equilateral triangles of two different sizes. Hence, we may assume that χ contains a monochromatic equilateral (a,a,a)-triangle ABC whose vertices avoid $\Delta(\chi)$. Assume that the three points A, B and C are black. Consider the configuration of eight points on Fig. 1. As mentioned in the proof of the first part of Lemma 1.3, every coloring of the five points D, A', B', C' and D' yields a monochromatic (a,b,c)-triangle. Furthermore, we may assume that the eight points all avoid $\Delta(\chi)$, otherwise we might shift the configuration slightly to move the points away from the boundary, without changing the color of ABC (recall that A, B and C belong to the interior of the black color). This concludes the proof. ∎

4. Concluding remarks

The Conjecture 1.2 remains open, despite the indirect support from our results, as well as from earlier research. The validity of this conjecture might depend on the particular choice of set-theoretical axioms. Such issues do not arise in this paper, since our proof techniques are very elementary. Unfortunately, these elementary techniques do not offer much hope for broad generalizations. It might nevertheless be possible to extend our results about polygonal colorings to a broader class of colorings, e.g., the colorings by monochromatic regions bounded by arbitrary Jordan curves. Such colorings have already been studied in the context of the related problem of the chromatic number of the plane (see [11]).

The zebra-like colorings provide a hitherto unknown example of colorings that avoid an equilateral triangle. We are not aware of any other examples of colorings avoiding a given triangle, but we do not dare to make any conjectures about the uniqueness of our construction, because our understanding of non-polygonal colorings is rather limited.

Acknowledgments

We appreciate the useful discussions with Zdeněk Dvořák, Jan Kratochvíl, Martin Tancer, Pavel Valtr and Tomáš Vyskočil.

References

[1] M. Bóna and G. Tóth: A Ramsey-type problem on right-angled triangles in space, *Discrete Math.* **250** (1996), 61–67.

[2] P. Erdős, R. L. Graham, P. Montgomery, B. L. Rothschild, J. Spencer and E. G. Straus: Euclidean Ramsey Theorems I, *Journal of Comb. Theory (A)* **14** (1973), 341–363.

[3] P. Erdős, R. L. Graham, P. Montgomery, B. L. Rothschild, J. Spencer and E. G. Straus: Euclidean Ramsey Theorems II, *Infinite and Finite Sets* **10** (1973), 529–557.

[4] P. Erdős, R. L. Graham, P. Montgomery, B. L. Rothschild, J. Spencer and E. G. Straus: Euclidean Ramsey Theorems III, *Infinite and Finite Sets* **10** (1973), 559–583.

[5] P. Frankl and V. Rödl: A Partition Property of simplices in Euclidean space, *Journal of the Amer. Math. Soc.* **3(1)** (1990), 1–7.

[6] R. L. Graham: Recent trends in Euclidean Ramsey theory, *Discrete Math.* **136** (1994), 119–127.

[7] I. Kříž: Permutation groups in Euclidean Ramsey theory, *Proc. Amer. Math. Soc.* **112** (1991), 899–907.

[8] I. Kříž: All trapezoids are Ramsey, *Discrete Math.* **108** (1992), 59–62.

[9] J. Matoušek and V. Rödl: On Ramsey Sets in Spheres, *Journal of Comb. Theory (A)* **70** (1995), 30–44.

[10] L. E. Shader: All right triangles are Ramsey in E^2!, *Journal of Comb. Theory (A)* **20** (1979), 385–389.

[11] D. R. Woodall: Distances realized by sets covering the plane, *Journal of Comb. Theory (A)* **14** (1973), 187–200.

Vít Jelínek

*Institute for Theoretical Computer Science (ITI)**
and Department of Applied Mathematics (KAM)†
Charles University
Faculty of Mathematics and Physics
Malostranské nám. 2/25
118 00, Prague
Czech Republic
jelinek@kam.mff.cuni.cz

Jan Kynčl, Rudolf Stolař, Tomáš Valla

Department of Applied Mathematics (KAM)†
Charles University
Faculty of Mathematics and Physics
Malostranské nám. 2/25
118 00, Prague
Czech Republic
jankyncl@centrum.cz, ruda@kam.mff.cuni.cz, valla@kam.mff.cuni.cz

* ITI is supported by project 1M0021620808 of the Czech Ministry of Education.
† KAM is supported by project MSM0021620838 of the Czech Ministry of Education.

Appendix E

Slope Number of Planar Partial 3-Trees

Graphs and Combinatorics (2013) 29:981–1005
DOI 10.1007/s00373-012-1157-z

ORIGINAL PAPER

The Planar Slope Number of Planar Partial 3-Trees of Bounded Degree

Vít Jelínek · Eva Jelínková · Jan Kratochvíl · Bernard Lidický · Marek Tesař · Tomáš Vyskočil

Received: 18 October 2010 / Revised: 29 February 2012 / Published online: 21 March 2012

Abstract It is known that every planar graph has a planar embedding where edges are represented by non-crossing straight-line segments. We study the planar slope number, i.e., the minimum number of distinct edge-slopes in such a drawing of a planar graph with maximum degree Δ. We show that the planar slope number of every planar partial 3-tree and also every plane partial 3-tree is at most $O(\Delta^5)$. In particular, we answer the question of Dujmović et al. (Comput Geom 38(3):194–212, 2007) whether there is a function f such that plane maximal outerplanar graphs can be drawn using at most $f(\Delta)$ slopes.

Supported by GraDR EUROGIGA project GIG/11/E023, by project 1M0021620838 of the Czech Ministry of Education, and by Grant 1M0545 of the Czech Ministry of Education.

V. Jelínek ( ) · E. Jelínková · J. Kratochvíl · B. Lidický · M. Tesař · T. Vyskočil
Department of Applied Mathematics, Charles University, Malostranské náměstí 25, Prague, Czech Republic
e-mail: jelinek@kam.mff.cuni.cz

E. Jelínková
e-mail: eva@kam.mff.cuni.cz

B. Lidický
e-mail: bernard@kam.mff.cuni.cz

M. Tesař
e-mail: tesulo@kam.mff.cuni.cz

J. Kratochvíl · T. Vyskočil
Institute for Theoretical Computer Science, Charles University, Malostranské náměstí 25, Prague, Czech Republic
e-mail: honza@kam.mff.cuni.cz

T. Vyskočil
e-mail: whisky@kam.mff.cuni.cz

Keywords Graph drawing · Planar graphs · Slopes · Planar slope number

Mathematics Subject Classification 68R10 · 05C10 · 05C62

1 Introduction

The *slope number* of a graph G was introduced by Wade and Chu [13]. It is defined as the minimum number of distinct edge-slopes in a straight-line drawing of G. Clearly, the slope number of G is at most the number of edges of G, and it is at least half of the maximum degree Δ of G.

Dujmović et al. [2] asked whether there was a function f such that each graph with maximum degree Δ could be drawn using at most $f(\Delta)$ slopes. In general, the answer is *no* due to a result of Barát et al. [1]. Later, Pach and Pálvölgyi [12] and Dujmović et al. [3] proved that for every $\Delta \geq 5$, there are graphs of maximum degree Δ that need an arbitrarily large number of slopes.

On the other hand, Keszegh et al. [7] proved that every graph of maximum degree three can be drawn using at most five slopes, and if we additionally assume that the graph is connected and has at least one vertex of degree less than three then four slopes suffice. Mukkamala and Szegedy [10] have shown that four slopes also suffice for every connected cubic graph. It was further strengthened by Mukkamala and Pálvölgyi [11] by showing that four basic slopes $\{0, \pi/4, \pi/2, 3\pi/4\}$ suffice for every cubic graph. Dujmović et al. [3] give a number of bounds in terms of the maximum degree: for interval graphs, cocomparability graphs, or AT-free graphs. All the results mentioned so far are related to straight-line drawings which are not necessarily non-crossing.

It is known that every planar graph G can be drawn so that edges of G are represented by non-crossing segments [6]. We call such a planar drawing a *straight-line embedding* of G. In this paper, we examine the minimum number of slopes in a straight-line embedding of a planar graph.

In this paper, we make the (standard) distinction between *planar graphs*, which are graphs that admit a plane embedding, and *plane graphs*, which are graphs accompanied with a fixed prescribed combinatorial embedding, i.e., a prescribed face structure, including a prescribed outer face. Accordingly, we distinguish between the *planar slope number* of a planar graph G, which is the smallest number of slopes needed to construct any straight-line embedding of G, as opposed to the *plane slope number* of a plane graph G, which is the smallest number of slopes needed to realize the prescribed combinatorial embedding of G as a straight-line embedding.

The research of slope parameters related to plane embedding was initiated by Dujmović et al. [2]. In [4], there are numerous results for the plane slope number of various classes of graphs. For instance, it is proved that every plane 3-tree can be drawn using at most $2n$ slopes, where n is its number of vertices. It is also shown that every 3-connected plane cubic graph can be drawn using three slopes, except three edges on the outer face.

Recently, Keszegh, Pach and Pálvölgyi [8] have shown that any plane graph of maximum degree Δ can be drawn with $2^{O(\Delta)}$ slopes. Their argument is based on a representation of planar graphs by touching disks.

In this paper, we study both the plane slope number and the planar slope number. The lower bounds of [1,3,12] for bounded-degree graphs do not apply to our case, because the constructed graphs with large slope numbers are not planar. Moreover, the upper bounds of [7,10] give drawings that contain crossings even for planar graphs.

For a fixed $k \in \mathbb{N}$, a *k-tree* is defined recursively as follows. A complete graph on k vertices is a k-tree. If G is a k-tree and K is a k-clique of G, then the graph formed by adding a new vertex to G and making it adjacent to all vertices of K is also a k-tree. A subgraph of a k-tree is called a *partial k-tree*.

We present several upper bounds on the plane and planar slope number in terms of the maximum degree Δ. The most general result of this paper is the following theorem, which deals with plane partial 3-trees.

Theorem 1.1 *The plane slope number of any plane partial 3-tree with maximum degree Δ is at most $O(\Delta^5)$.*

Note that the above theorem implies that the planar slope number of any planar partial 3-tree is also at most $O(\Delta^5)$.

Since every outerplanar graph is also a partial 3-tree, the result above answers a question of Dujmović et al. [4], who asked whether a plane maximal outerplanar graph can be drawn using at most $f(\Delta)$ slopes.

Unlike the results of Keszegh, Pach and Pálvölgyi [8], our arguments are only applicable to a restricted class of planar graphs. On the other hand, our bound is polynomial in Δ rather than exponential, and moreover, our proof is constructive.

A weaker version of our results has been announced in an extended abstract that was presented at Graph Drawing 2009 [5].

2 Preliminaries

Let us introduce some basic terminology and notation that will be used throughout this paper.

Let s be a segment in the plane. The smallest angle $\alpha \in [0, \pi)$ such that s can be made horizontal by a clockwise rotation by α, is called the *slope of s*. The *directed slope* of a directed segment is an angle $\alpha' \in [0, 2\pi)$ defined analogously.

A plane graph is called a *near triangulation* if all its faces, except possibly the outer face, are triangles.

3 Plane Partial 3-Trees

In this section we present the proof of Theorem 1.1. We start with some observations about the structure of plane 3-trees. Throughout this section, we assume that Δ is a fixed integer.

It is known that any plane 3-tree can be generated from a triangle by a sequence of vertex-insertions into inner faces. Here, a *vertex-insertion* is an operation that consists of creating a new vertex in the interior of a face, and then connecting the new vertex to all the three vertices of the surrounding face, thus subdividing the face into three new faces.

For a plane partial 3-tree G we define the *level* of a vertex v as the smallest integer k such that there is a set V_0 of k vertices of G with the property that v is on the outer face of the plane graph $G - V_0$. Let G be a plane partial 3-tree. An edge of G is called *balanced* if it connects two vertices of the same level of G. An edge that is not balanced is called *tilted*. Similarly, a face of G whose all vertices belong to the same level is called balanced, and any other face is called tilted. In a plane 3-tree, the level of a vertex v can also be equivalently defined as the length of the shortest path from v to a vertex on the outer face. However, this definition cannot be used for plane partial 3-trees.

Note that whenever we insert a new vertex v into an inner face of a 3-tree, the level of v is one higher than the minimum level of its three neighbors; note also that the level of all the remaining vertices of the 3-tree is not affected by the insertion of a new vertex.

Let u, v be a pair of vertices forming an edge. A *bubble* over uv is an outerplanar plane near triangulation that contains the edge uv on the boundary of the outer face. The edge uv is called the *root* of the bubble. A *trivial bubble* is a bubble that has no other edge apart from the root edge. A *double bubble* over uv is a union of two bubbles over uv which have only u and v in common and are attached to uv from its opposite sides. A *leg* is a graph L created from a path P by adding a double bubble over every edge of P. The path P is called the *spine of L* and the endpoints of P are also referred to as the endpoints of the leg. Note that a single vertex is also considered to form a leg.

A *tripod* is a union of three legs which share a common endpoint. A *spine* of a tripod is the union of the spines of its legs. Observe that a tripod is an outerplanar graph. The vertex that is shared by all the three legs of a tripod is called *the central vertex*.

Let G be a near triangulation, let Φ be an inner face of G. Let T be a tripod with three legs X, Y, Z and a central vertex c. An *insertion of tripod T into the face Φ* is the operation performed as follows. First, insert the central vertex c into the interior of Φ and connect it by edges to the three vertices of Φ. This subdivides Φ into three subfaces. Extend c into an embedding of the whole tripod T, by embedding a single leg of the tripod into the interior of each of the three subfaces. Next, connect every non-central vertex of the spine of the tripod to the two vertices of Φ that share a face with the corresponding leg. Finally, connect each non-spine vertex v of the tripod to the single vertex of Φ that shares a face with v. See Fig. 1. Observe that the graph obtained by a tripod insertion into Φ is again a near triangulation.

Lemma 3.1 *Let G be a graph. The following statements are equivalent:*

1. *G is a plane 3-tree, i.e., G can be created from a triangle by a sequence of vertex insertions into inner faces.*
2. *G can be created from a triangle by a sequence of tripod insertions into inner faces.*
3. *G can be created from a triangle by a sequence of tripod insertions into balanced inner faces.*

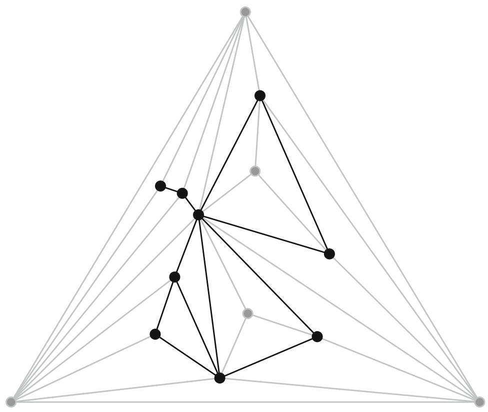


Fig. 1 An example of a tripod consisting of vertices of level 1 in a plane 3-tree

Proof Clearly, (3) implies (2).

To observe that (2) implies (1), it suffices to notice that a tripod insertion into a face Φ can be simulated by a sequence of vertex insertions: first insert the central vertex of a tripod into Φ, then insert the vertices of the spine into the resulting subfaces, and then create each bubble by inserting vertices into the face that contains the root of the bubble and its subsequent subfaces.

To show that (1) implies (3), proceed by induction on the number of levels in G. If G only has vertices of level 0, then it consists of a single triangle and there is nothing to prove. Assume now that G is a graph that contains vertices of $k > 0$ distinct levels, and assume that any 3-tree with fewer levels can be generated by a sequence of balanced tripod insertions by induction.

We will show that the vertices of level k induce in G a subgraph whose every connected component is a tripod, and that each of these tripods is inserted inside a triangle whose vertices have level $k-1$.

Let C be a connected component of the subgraph induced in G by the vertices of level k. Let $v_1, v_2, \dots, v_m$ be the vertices of C, in the order in which they were inserted when G was created by a sequence of vertex insertions. Let Φ be the triangle into which the vertex v_1 was inserted, and let x, y and z be the vertices of Φ. Necessarily, all three of these vertices have level $k-1$. Each of the vertices $v_2, \dots, v_m$ must have been inserted into the interior of Φ, and each of them must have been inserted into a face that contained at least one of the three vertices of Φ.

Note that at each point after the insertion of v_1, there are exactly three faces inside Φ that contain a pair of vertices of Φ; each of these three faces is incident to an edge of Φ. Whenever a vertex v_i is inserted into such a face, the subgraph induced by vertices of level k grows by a single edge. These edges form a union of three paths that share the vertex v_1 as their common endpoint.

On the other hand, when a vertex v_i is inserted into a face formed by a single vertex of Φ and a pair of previously inserted vertices v_j, v_ℓ, then the graph induced by vertices of level k grows by two new edges $v_i v_j$ and $v_i v_\ell$, as well as a new triangular face with vertices v_i, v_j, v_ℓ.

With these observations, it is easy to check (e.g., by induction on i) that for every $i \geq 1$, the subgraph of G induced by the vertices $v_1, \dots, v_i$ is a tripod inserted into Φ.

From this fact, it follows that the whole graph G may have been created by a sequence of tripod insertions into balanced faces. □

Note that when we insert a tripod into a balanced face, all the vertices of the tripod will have the same level (which will be one higher than the level of the face into which we insert the tripod). In particular, each balanced face we create by this insertion is an inner face of the inserted tripod.

Recall that a plane partial 3-tree is a plane graph that is a subgraph of a 3-tree. Kratochvíl and Vaner [9] have shown that every plane partial 3-tree G is in fact a subgraph of a plane 3-tree. Furthermore, if a plane partial 3-tree G has at least three vertices, it is in fact a spanning subgraph of a plane 3-tree, i.e., it can be extended into a plane 3-tree by only adding edges.

Unfortunately, the plane 3-tree that contains a plane partial 3-tree G may in general require arbitrarily large vertex-degrees, even if the maximum degree of G is bounded. Thus, the result of Kratochvíl and Vaner does not allow us to directly simplify the problem to drawing plane 3-trees.

To overcome this difficulty, we introduce the notion of 'plane semi-partial 3-tree', which can be seen as an intermediate concept between plane 3-trees and plane partial 3-trees.

Definition 3.1 A graph G is called *a plane semi-partial 3-tree* if G is obtained from a plane 3-tree H by erasing some of the tilted edges of H.

Our goal is to prove that every plane partial 3-tree of maximum degree Δ can be drawn with at most $O(\Delta^5)$ slopes. We obtain this result as a direct consequence of two main propositions, stated below.

Proposition 3.1 *Any connected plane partial 3-tree of maximum degree Δ is a subgraph of a plane semi-partial 3-tree of maximum degree at most 37Δ.*

Proposition 3.2 *For every Δ there is a set S of at most $O(\Delta^5)$ slopes with the property that any plane semi-partial 3-tree of maximum degree Δ has a straight-line embedding whose edge-slopes all belong to S.*

We begin by proving Proposition 3.1.

3.1 Proof of Proposition 3.1

We begin by a simple lemma, which shows that the deletion of tilted edges from a plane 3-tree does not affect the level of vertices.

Lemma 3.2 *Let $H = (V, E)$ be a plane 3-tree, let T be a set of tilted edges of H, let $G = (V, E \setminus T)$ be a semi-partial 3-tree. Let v be a vertex of level k with respect to H. Then v has level k in G as well.*

Proof Fix a vertex v of level k in H. Of course, the deletion of an edge may only decrease the level of a vertex, so v has level at most k in G. On the other hand, it follows from Lemma 3.1 that every vertex of level k in H is separated from the outer face by k nested triangles $C_0, C_1, \dots C_{k-1}$, where C_i is a triangle formed by balanced edges that belong to level i. Since every balanced edge of H belongs to G as well, we know that all the triangles $C_0, C_1, \dots C_{k-1}$ belong to G, showing that v has level at least k. It follows that the level of v is preserved by the deletion of tilted edges. □

Let $G = (V, E)$ be a plane semi-partial 3-tree obtained from a plane 3-tree $H = (V, E')$ by the deletion of several tilted edges. As a consequence of the previous lemma, we see that an edge $e \in E$ is tilted in G if and only if it is tilted in H.

Assume now that F is a connected plane partial 3-tree with maximum degree $\Delta \geq 1$ and at least three vertices. Our goal is to show that there is a plane semi-partial 3-tree G with maximum degree at most 37Δ that contains F as a spanning subgraph. The following definition introduces the key notion of our proof.

Definition 3.2 Let F be a connected plane partial 3-tree with maximum degree Δ, and let k be an integer. We say that a 3-tree H *correctly covers F up to level k*, if the following conditions are satisfied:

- F is a spanning subgraph of H.
- Let $V^{\leq k}$ denote the set of vertices that have level at most k in H. For every vertex $v \in V^{\leq k}$ there are at most 36Δ balanced edges of H that are incident to v.

Furthermore, we say that H *correctly covers F at all levels* if, for any k, H correctly covers F up to level k.

As mentioned before, Kratochvíl and Vaner [9] have shown that every plane partial 3-tree F is a spanning subgraph of a plane 3-tree H. Note that such a 3-tree H correctly covers F up to level 0, because every vertex at level 0 is adjacent to two balanced edges.

Our proof of Proposition 3.1 is based on the following lemma.

Lemma 3.3 *For every connected partial 3-tree F there is a 3-tree H that correctly covers F at all levels.*

Before we prove the lemma, let us show how it implies Proposition 3.1.

Proof of Proposition 3.1 from Lemma 3.3 Let F be a plane partial 3-tree of maximum degree Δ, and let H be the 3-tree that correctly covers F at all levels. Define a semi-partial 3-tree G which is obtained from H by erasing all the tilted edges of H that do not belong to F. By construction, G is a semi-partial 3-tree that contains F as a subgraph. Moreover, every vertex of G is adjacent to at most Δ tilted edges and at most 36Δ balanced edges, so G has maximum degree at most 37Δ. □

Let us now turn to the proof of Lemma 3.3.

Proof Let F be a partial 3-tree with maximum degree Δ, and assume for contradiction that there is no graph H that would correctly cover F. Let k be the largest integer such that there is a graph H that correctly covers F up to level k. We have seen that $k \geq 0$. On the other hand, we clearly have $k < |V(F)|$. Thus, k is well defined.

Fix a graph H correctly covering F up to level k. By our assumption, H has vertices of level greater than k. We will now define a 3-tree H' that correctly covers F up to level $k + 1$, which contradicts the maximality of k.

Note that it is sufficient to ensure that H' is constructed by a sequence of balanced tripod insertions in which all the tripods inserted at level at most $k + 1$ have degrees bounded by 36Δ.

We construct H' in such a way that it coincides with H on vertices of level at most k; more precisely, if u and v are two vertices of level at most k in H, then u and v are connected by an edge of H' if and only if they are connected by an edge of H. Notice that this property guarantees that the vertices at level at most k in H are at the same level in H' as in H. Let $H^{\le k}$ be the subgraph of H induced by the vertices of level at most k. $H^{\le k}$ is a 3-tree.

Let Φ be a balanced face of $H^{\le k}$ formed by vertices at level k which contains at least one vertex of H at level $k + 1$ in its interior. Note that at least one such face exists, since we assumed that at least one vertex has level greater than k in H. For any such face Φ, we will modify the sequence of tripod-insertions performed inside Φ, such that the tripod inserted into this face has maximum degree at most 36Δ, while the modified graph will still contain F as a subgraph. By doing this modification inside every nonempty balanced face at level k, we will eventually obtain a graph H' that correctly covers F up to level $k + 1$.

Fix Φ to be a balanced face at level k with nonempty interior. Let $T \subset H$ be the tripod that has been inserted into Φ during the construction of H. Let V_T and E_T be the vertices and the edges of T. We will now define a modified tripod T' on the vertex set V_T, satisfying the required degree bound. We will then show that the sequence of tripod insertions that have been performed inside T during the construction of H can be transformed into a sequence of tripod insertions inside T', where the new sequence of insertions yields a graph H' that contains F as a subgraph.

We define T' by the following rules.

1. All the edges of T that belong to F are also in T'.
2. All the edges of T that belong to the boundary of the outer face of T also belong to T'. These edges form the boundary of the outer face of T'.
3. All the edges that form the spine of T also belong to T' and they form its spine.
4. Let Ψ be an internal face of the tripod T. Let u, v and w be the three vertices of Ψ. Assume that both u and v are connected by an edge of F to a vertex in the interior of Ψ (not necessarily both of them to the same vertex). In such case, add the edge uv to T'.
5. Let T'_0 be the graph formed by all the edges added to T' by the previous four rules. Note that T'_0 is an outerplanar graph with the same outer face as T. However, not all the inner faces of T'_0 are necessarily triangles, so T'_0 is not necessarily a tripod. Assume that T'_0 has an inner face with more than three vertices, and that $v_0, v_1, \ldots, v_r$ are the vertices of this face, listed in cyclic order. We form the path $v_1, v_r, v_2, v_{r-1}, v_3, v_{r-2}, \ldots$ whose edges triangulate the face of T'_0. We add all the edges of this path into T'. We do this for every internal face of T'_0 that has more than three vertices. The resulting graph T' is clearly a tripod.

Let us now argue that the tripod T' has maximum degree at most 36Δ. Let $v \in V_T$ be any vertex of this tripod. Let us estimate $\deg_{T'}(v)$, by counting the edges adjacent to v that were added to T' by the rules above. Clearly, there are at most Δ such edges that were added by the first rule, and at most nine such edges that were added by the second and third rule.

We claim that there are at most 2Δ edges incident with v added by the fourth rule. To see this, notice that if $e = uv$ is an edge added by this rule, then at least one of the two faces of T that are incident to e must contain in its interior an edge e' of F that is incident to v. In such situation, we say that e' is *responsible* for the insertion of e into T'. Clearly, an edge of F may be responsible for the insertion of at most two edges incident with v. Since v has degree at most Δ in F, this shows that at most 2Δ edges incident with v are added to T'_0 by the fourth rule. Consequently, T'_0 has maximum degree at most $3\Delta + 9$.

To estimate the number of edges added to T' by the fifth rule, it is sufficient to observe that in every internal face of T'_0 whose boundary contains v there are at most two edges of T' incident to v added by the fifth rule. Thus, $\Delta(T') \le 3\Delta(T'0) \le 9\Delta + 27 \le 36\Delta$, as claimed.

Having thus defined the tripod T', we modify the graph H as follows. We remove all the vertices appearing in the interior of the face Φ of $H^{\le k}$; that is, we remove the tripod T as well as all the vertices inserted inside T. Instead, as a first step towards the construction of H', we insert T' inside Φ.

To finish the construction of H', we need to insert the vertices of level greater than $k + 1$ into the faces of T', so that the resulting graph contains F as a subgraph. We perform this insertion separately inside every face of T'_0. Note that T'_0 is a subgraph of T as well as a subgraph of T', and that each internal face of T'_0 is a union of several faces of T'. Let Ψ be a face of T'_0. If Ψ is a triangle, then Ψ is in fact a face of T' as well as a face of T. If T contains a subgraph H_Ψ inside Ψ, we define H' to contain the same subgraph inside Ψ as well. Since H_Ψ has been created by a sequence of tripod insertions inside Ψ, we can perform the same sequence of tripod insertion again inside the same face during the construction of H'.

Assume now that Ψ is not a triangle. In the graph H, the face Ψ is subdivided into a collection of triangular faces $\Psi_1, \Psi_2, \dots, \Psi_k$. Let H_i be the subgraph of H appearing inside the face Ψ_i in H. We know that each H_i is a result of a sequence of tripod insertions.

Let us use the following terminology: if there is an edge of F that connects a vertex of H_i to a vertex v on the boundary of Ψ, we say that H_i is *adjacent* to v. Since the graph F is connected, each nonempty graph H_i must be adjacent to at least one vertex on the boundary of Ψ. Observe that if H_i is adjacent to two distinct vertices u and v on the boundary of Ψ, then the edge that connects u and v must belong to T'_0 by the fourth rule in the construction of T'. In particular, u and v appear consecutively on the boundary of Ψ. This also shows that H_i cannot be adjacent to three distinct vertices of Ψ, since we assumed that Ψ is not a triangle.

Consider now the tripod T'. In this tripod, the face Ψ is triangulated into a collection of faces $\Psi'_1, \Psi'_2, \dots, \Psi'_k$. Each of these triangular faces has at least one edge of T'_0 on its boundary. We will insert the graphs $H_1, H_2, \dots, H_k$ into these faces, by performing for each H_i a sequence of tripod insertions which generates H_i inside one of the faces $\Psi'_1, \Psi'_2, \ldots, \Psi'_k$.

To ensure that the resulting graph will contain F as a subgraph, it suffices to guarantee that whenever H_i is adjacent to a vertex $v \in \Psi$, it will be inserted into a face Ψ'_j that contains v on its boundary. Such a face always exists, since each H_i is adjacent to at most two vertices of Ψ, and if it is adjacent to two vertices u, v, then the two vertices must be connected by an edge on the boundary of Ψ, which implies that there is a face Ψ'_j that contains both u and v on its boundary.

It may happen that two distinct graphs H_i and H_j need to be inserted into the same face Ψ'_ℓ. In such case, the first graph is inserted directly into Ψ'_ℓ, thus partitioning it into several smaller triangular subfaces, while all subsequent graphs that need to be inserted into Ψ'_ℓ are inserted into an appropriately chosen subface of Ψ'_ℓ. This subface need not be balanced. We choose this subface in such a way that we preserve the cyclic order of edges of F around every vertex v on the boundary of Ψ.

After we perform the construction above inside every face Ψ of T'_0, we obtain a plane 3-tree H' that correctly covers F up to level $k + 1$. This completes the proof of the lemma. □

3.2 Proof of Proposition 3.2

To complete the proof of our main result, it remains to show that every plane semi-partial 3-tree of bounded maximum degree has a straight-line embedding with a bounded number of slopes.

We start with a brief overview of the construction. We will use the fact that a plane semi-partial 3-tree G can be decomposed into tripods formed by vertices of the same level, with each tripod T of level $k \geq 1$ being inserted into a triangle Φ formed by vertices of level $k - 1$. The triangle Φ is itself an inner face of a tripod of level $k - 1$.

The tripods appearing in this decomposition of G may be arbitrarily large. However, a tripod T of level $k \geq 1$ has only a bounded number of vertices that are adjacent to a vertex of the triangle Φ of level $k - 1$. These vertices of T will be called *relevant vertices*.

Given a tripod T in the decomposition of G, we will construct an embedding of T that only uses edge-slopes from a set of slopes S' and moreover, all the relevant vertices of T are embedded on points from a set of points P', where the sets S' and P' are independent of T and their size is polynomial in Δ.

We will then show that these embeddings of tripods (after a suitable scaling) can be nested into each other to provide the embedding of the whole graph G. We will argue that the number of edge-slopes in this embedding of G is bounded. This will follow from the fact that the balanced edges of G belong to a tripod and their slope belongs to S', while the slopes of the tilted edges only depend on the positions of the relevant vertices of a tripod T and on the shape of the triangle Φ surrounding T. Since the relevant vertices can only have a bounded number of positions, and the triangle Φ is formed by balanced edges and hence may have only a bounded number of shapes, we will conclude that the tilted edges may only determine a bounded number of slopes.

Let us now describe the construction in detail. We recall that Δ is a fixed constant throughout this section, and we let $\mathrm{ST}(\Delta)$ denote the set of plane semi-partial 3-trees of maximum degree at most Δ. Any graph $G \in \mathrm{ST}(\Delta)$ can be created by a sequence of *partial tripod insertions* into balanced faces, where a partial tripod insertion is defined in the same way as an ordinary tripod insertion, except that some of the tilted edges are omitted when the new tripod is inserted.

Choose a graph $G \in \mathrm{ST}(\Delta)$, and assume that T is a tripod that is used in the construction of G by a sequence of partial tripod insertions. Let $\{x, y, z\}$ be the triangle in G into which the tripod T has been inserted. We say that a vertex v of T is *relevant* if v is connected by an edge of G to at least one of the vertices x, y or z. Since each of the three vertices x, y and z has degree at most Δ, the tripod T has at most 3Δ relevant vertices. Let us further say that a bubble of T is relevant if it contains at least one relevant vertex. Since every vertex of T is contained in at most six bubbles, we see that T has at most 18Δ relevant bubbles.

We will use the term *labelled tripod of degree* Δ to denote a tripod T with maximum degree at most Δ, together with an associated set of at most 3Δ relevant vertices of T. Let $\mathrm{Tr}(\Delta)$ be the (infinite) set of all the labelled tripods of degree Δ. Similarly, a *labelled bubble of degree* Δ is a bubble of maximum degree at most Δ, together with a prescribed set of at most 3Δ relevant vertices. $\mathscr{B}(\Delta)$ denotes the set of all such labelled bubbles.

Let $\mathscr{E}_T$ be an embedding of a tripod in the plane, and let v be a vertex of $\mathscr{E}_T$. Let $\alpha \in \langle 0, 2\pi)$ be a directed slope. We say that the vertex v has *visibility in direction* α, if the ray starting in v and having direction α does not intersect $\mathscr{E}_T$ in any point except v.

Throughout the rest of this section, let ε denote the value $\pi/100$ (any sufficiently small integral fraction of π is suitable here).

Our proof of Proposition 3.2 is based on the following key lemma.

Lemma 3.4 (Tripod Drawing Lemma) *For every Δ there is a set of slopes S of size $O(\Delta^3)$, a set of points P of size $O(\Delta^2)$, and a set of triangles R of size $O(\Delta^3)$, such that every labelled tripod $T \in Tr(\Delta)$ has a straight-line embedding $\mathscr{E}_T$ with the following properties:*

1. *The slope of any edge in the embedding $\mathscr{E}_T$ belongs to S.*
2. *Each relevant vertex of $\mathscr{E}_T$ is embedded on a point from P.*
3. *Each internal face of $\mathscr{E}_T$ is homothetic to a triangle from R.*
4. *The central vertex of $\mathscr{E}_T$ is embedded in the origin of the plane.*
5. *Any vertex of $\mathscr{E}_T$ is embedded at a distance at most 1 from the origin.*
6. *Each spine of T is embedded on a single ray starting from the origin. The three rays containing the spines have directed slopes 0, $2\pi/3$ and $4\pi/3$. Let these three rays be denoted by r_1, r_2 and r_3, respectively.*
7. *Let $\widehat{r_i r_j}$ denote the closed convex region whose boundary is formed by the rays r_i and r_j. Any relevant vertex of $\mathscr{E}_T$ embedded in the region $\widehat{r_1 r_2}$ (or $\widehat{r_2 r_3}$, or $\widehat{r_1 r_3}$) has visibility in any direction from the set $\langle \varepsilon, 2\pi/3 - \varepsilon\rangle$ (or $\langle 2\pi/3 + \varepsilon, 4\pi/3 - \varepsilon\rangle$, or $\langle 4\pi/3 + \varepsilon, 2\pi - \varepsilon\rangle$, respectively).*

   Note that the three regions $\widehat{r_1 r_2}$, $\widehat{r_2 r_3}$ and $\widehat{r_1 r_3}$ are not disjoint. For instance, if a relevant vertex of T is embedded on the ray r_1, it belongs to both $\widehat{r_1 r_2}$ and $\widehat{r_1 r_3}$,

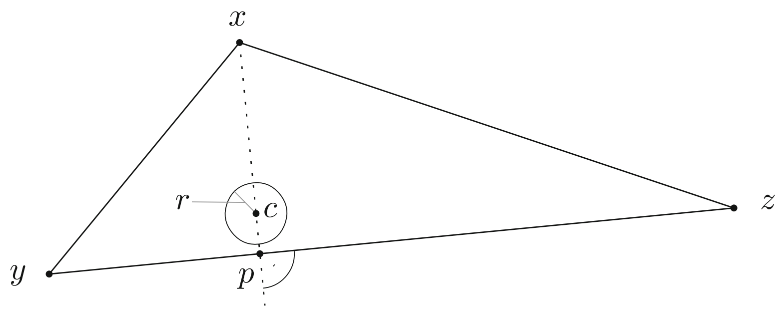


Fig. 2 Illustration of the proof of Proposition 3.2

and hence it must have visibility in any direction from the set $\langle\varepsilon, 2\pi/3 - \varepsilon\rangle \cup \langle 4\pi/3 + \varepsilon, 2\pi - \varepsilon\rangle$.

Before we prove Lemma 3.4, we show how the lemma implies Proposition 3.2.

Proof of Proposition 3.2 *from Lemma* 3.4 Let S be the set of slopes, P be the set of points and R be the set of triangles from Lemma 3.4. Let S' be the set of all the slopes that differ from a slope in S by an integer multiple of ε. Note that $|S'| \leq \frac{\pi}{\varepsilon}|S|$. Let P' be the (finite) set of points that can be obtained by rotating a point in P around the origin by an integral multiple of ε. Let R' be the (finite) set of triangles that is obtained by rotating the triangles in R by an integral multiple of ε.

We will show that any graph $G \in \mathrm{ST}(\Delta)$ has a straight-line embedding where the slopes of balanced edges belong to S' and the slopes of tilted edges also belong to a finite set which is independent of G.

Let T be a labelled tripod used in the construction of the graph G. Assume that T is inserted into a triangle formed by three vertices x, y, z (see Fig. 2). Let τ be the triangle formed by the three points x, y, z. Assume that the three vertices are embedded in the plane. Without loss of generality, assume that the triangle τ has acute angles by the vertices y and z, and the three vertices xyz appear in counterclockwise order around the boundary of τ. Thus the altitude of τ from the vertex x intersects the segment yz on a point p which is in the interior of the segment yz. Let η be the slope of the (directed) segment yz.

We can find a point c in the interior of the triangle τ, and a positive real number $r = r(\tau)$, such that for any point v at a distance at most r from c, the following holds:

1. v is in the interior of τ
2. the slope of the segment vx differs from the slope of the segment px (which is equal to $\eta + \pi/2$) by less than ε
3. the slope of the segment vy differs from the slope of the segment py (which is equal to $-\eta$) by less than ε
4. the slope of the segment vz differs from the slope of the segment pz (which is equal to η) by less than ε

Indeed, it suffices to choose c sufficiently close to the point p and set r sufficiently small, and all the above conditions will be satisfied.

Consider now the embedding $\mathscr{E}_T$ of T. Place the center of the tripod on the point c, and scale the whole embedding by the factor r, so that it fits inside the triangle τ.

In view of the four conditions above, and in view of the seventh part of Lemma 3.4, it is not difficult to observe that we may rotate the (scaled) embedding of T around the point c by an integral multiple of ε in such a way that every relevant vertex $v \in T$ has visibility towards all its neighbors among the three vertices x, y, z. Thus, we are able to embed all the necessary tilted edges of G between xyz and T as straight line segments.

Note that in our embedding, all the balanced edges of T have slopes from the set S', and all its internal faces are homothetic to the triangles from the set R'. Furthermore, any tilted edge has one endpoint in the set $\{x, y, z\}$ and another endpoint in the set $c + rP'$ (the set P' scaled r-fold and translated in such a way that the origin is moved to c). Hence any labelled tripod $T \in \mathrm{Tr}(\Delta)$ can be inserted inside the triangle xyz in such a way that the slopes of the edges always belong to the same finite set which depends on the triangle xyz but not on the tripod T. Note that the triangle xyz may be arbitrarily thin, in particular it can have inner angles smaller than ε.

Let us now show how the above construction yields an embedding of the whole graph G. For every such triangle $\tau \in R'$, fix the point $c = c(\tau)$ and the radius $r = r(\tau)$ from the above construction. Any scaled and translated copy of τ will have the values of c and r scaled and translated accordingly.

We now embed the graph G recursively, by embedding the outer face as an arbitrary triangle from R', and then recursively embedding each tripod into the appropriate face by the procedure described above. Since we only insert tripods into balanced faces, it is easily seen that every tripod is being embedded inside a triangle of R'.

Overall, the construction uses at most $|S'| = O(\Delta^3)$ distinct slopes for the balanced edges, and at most $|R'||P'| = O(\Delta^5)$ distinct slopes for the tilted edges. The total number of slopes is then $O(\Delta^5)$, as claimed. □

In the rest of this section, we prove the Tripod Drawing Lemma. Let T be a labelled tripod and let B be a bubble of T. Recall that the root edge of B is the edge that belongs to a spine of T. Note that the same root edge is shared by two bubbles of T. Recall also that a bubble is called trivial if it only has two vertices.

We now introduce some terminology that will be convenient for our description of the structure of a given bubble.

Definition 3.3 Let B be a nontrivial bubble in a tripod T. The unique internal face of B adjacent to its root edge will be called *the root face* of B. The *dual* of a bubble B is the rooted binary tree $\widehat{B}$ whose nodes correspond bijectively to the internal faces of B, and two nodes are adjacent if and only if the corresponding faces of B share an edge. The root of the tree $\widehat{B}$ is the node that represents the root face of B.

When dealing with the internal faces of B, we will employ the usual terminology of rooted trees; for instance, we say that a face Φ is the parent (or child) of a face Ψ if the node representing Φ in $\widehat{B}$ is the parent (or child) of the node representing Ψ. For every internal face Φ of B, the three edges that form the boundary of Φ will be called *the top edge, the left edge* and *the right edge*, where the top edge is the edge that Φ shares with its parent face (or the root edge, if Φ is the root face), while left and right edges are defined in such a way that the top, left, and right edge form a counterclockwise sequence on the boundary of Φ. With this convention, we may speak of a left child face or right child face of Φ without any ambiguity. Our terminology is motivated by the usual convention of embedding rooted binary trees with their root on the top, and the parent, the left child and the right child appearing in counterclockwise order around every node of the tree. Furthermore, for a given face Φ, the *bottom vertex* of Φ is the common vertex of the left edge and right edge of Φ.

Let us explicitly state the following simple fact which directly follows from our definitions.

Observation 3.1 *Let $\Phi_1, \Phi_2, \ldots, \Phi_k$ be a sequence of internal faces of a bubble B, such that for any $j < k$, Φ_{j+1} is the left child of Φ_j. Then all the faces $\Phi_1, \ldots, \Phi_k$ share a common vertex. In particular, if B has maximum degree Δ, then $k < \Delta$. An analogous observation holds for right children as well.*

We now describe an approach that allows us to embed an arbitrary bubble with maximum degree Δ inside a bounded area using a bounded number of slopes.

Lemma 3.5 *Let xyz be an equilateral triangle with vertex coordinates $x = (0, 0)$, $y = (1, 0)$ and $z = (1/2, -\sqrt{3}/2)$. Fix two sequences of slopes $\alpha_1, \alpha_2, \ldots, \alpha_{\Delta-1}$ and $\beta_1, \beta_2, \ldots, \beta_{\Delta-1}$, with $0 > \alpha_1 > \alpha_2 > \cdots b > \alpha_{\Delta-1} > -\pi/3$ and $0 < \beta_1 < \beta_2 < \cdots b < \beta_{\Delta-1} < \pi/3$. Let S be the set of $2\Delta - 1$ slopes $\{0\} \cup \{\alpha_1, \alpha_2, \ldots, \alpha_{\Delta-1}\} \cup \{\beta_1, \beta_2, \ldots, \beta_{\Delta-1}\}$. Let B be a bubble of maximum degree Δ. Then B has a straight line embedding $\mathscr{E}_B$ inside xyz that only uses the slopes from the set S, the root edge of $\mathscr{E}_B$ corresponds to the segment xy, and moreover the triangular faces of $\mathscr{E}_B$ form at most $2\Delta - 3$ distinct triangles up to homothetic equivalence.*

Proof Proceed by induction on the size of B. If B is trivial, the statement holds. Assume now that B is a nontrivial bubble. Let Φ_0 be the root edge of$B$. See Fig. 3.

Define the maximal sequence of faces $\Phi_1, \Phi_2, \ldots, \Phi_\ell$ in such a way that Φ_{i+1} is the left child of Φ_i, with Φ_1 being the left child of the root edge Φ_0. The maximality of the sequence means that Φ_ℓ has no left child. Symmetrically, define a maximal sequence of faces $\Psi_1, \ldots, \Psi_r$ such that Ψ_1 is the right child of Φ_0, and Ψ_{i+1} is the right child of Ψ_i. By Observation 3.1, we know that $\ell < \Delta - 1$ and $r < \Delta - 1$.

Let (p, α) denote the ray starting at a point p and heading in direction α.

Let B be an arbitrary bubble. Let v_1 be the intersection of the rays (x, α_1) and (y, β_1). The root face Φ_0 will be embedded as the triangle xyv_1. Define the points $v_2, \ldots, v_{\ell+1}$ by specifying v_i as the intersection of (x, α_i) and (v_1, π). The face Φ_i is then embedded as the triangle xv_iv_{i+1}. Similarly, define points $w_2, \ldots, w_{r+1}$ where w_i is the intersection of (y, β_i) with $(v_1, 0)$. Then Ψ_1 is embedded as the triangle yv_1w_2, while for $k > 1$ we embed Ψ_k as the triangle yw_kw_{k+1}.

Note that when we remove the two vertices incident to the root edge from the bubble B, the remaining edges and vertices form a union of $\ell + r$ bubbles $B_1 \cup \cdots \cup B_\ell \cup B'_1 \cup \cdots \cup B'_\ell$, where B_i is a bubble whose root edge is the right edge of Φ_i while B'_j is rooted at the left edge of Ψ_j. Using induction, we know that each B_i has a straight line embedding inside the equilateral triangle whose top edge is the horizontal segment v_iv_{i+1} (and symmetrically for B'_j).

This completes the proof. □

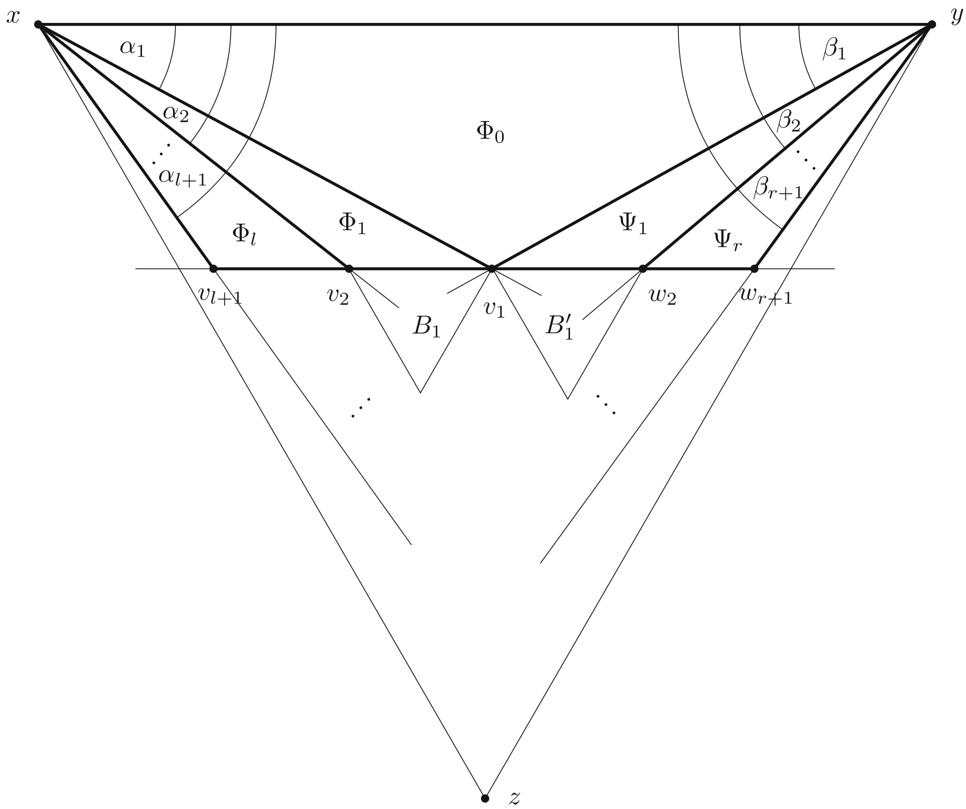


Fig. 3 Illustration of the proof of Lemma 3.5

Corollary 3.1 *Let xyz be an arbitrary triangle and B a bubble of maximum degree Δ. There are sets S of $2\Delta - 1$ slopes and R of $2\Delta - 3$ triangles that depend on xyz but not on B, such that B can be embedded inside xyz using only slopes from S and triangles from R for triangular faces, in such a way that the root edge of B coincides with the segment xy.*

Proof This follows from Lemma 3.5, using the fact that for any triangle there is an affine transform that maps it to an equilateral triangle, and that affine transforms preserve the number of distinct slopes used in a straight-line embedding. □

The construction from Lemma 3.5 can be applied to embed all the irrelevant bubbles of a given labelled tripod T. Unfortunately, the construction of Lemma 3.5 is not suitable for the embedding of relevant bubbles, because it provides no control about the position of the relevant vertices. Indeed, inside the triangle xyz of the previous lemma, there are infinitely many points where a vertex may be embedded by the construction described in the proof of the lemma. Thus, we can give no upper bound on the number of potential embeddings of relevant vertices.

For this reason, we now describe a more complicated embedding procedure, which allows us to control the position of the relevant vertices. We first need some auxiliary definitions.

Definition 3.4 An *adder* A is a bubble with a root edge h and another edge $t \neq h$, such that the dual tree of A is a path, and the edge t is an external edge adjacent to the single leaf face of A. See Fig. 4. The edges h and t are called *head* and *tail* of the

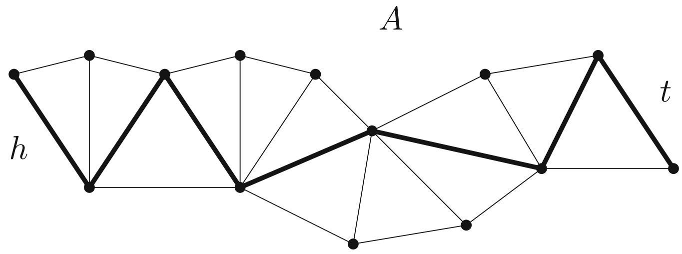


Fig. 4 An adder. The *bold edges* form the zigzag path

adder. It is easy to see that every adder contains a unique path Z whose first edge is h, its last edge is t and no other edge of Z belongs to the outer face of A. The path Z will be called the *zigzag path* of the adder A. The *length* of the adder is defined to be the number of edges of its zigzag path. By definition, each adder has length at least 2. An adder of length 2 will be called *degenerate*.

We will now show that adders of bounded degree can be embedded inside a prescribed quadrilateral using a bounded number of slopes and triangles.

Lemma 3.6 *For every convex quadrilateral $Q = abcd$ and for every Δ there is a set S of $O(\Delta)$ slopes, a set $S_0 \subseteq S$ of $O(1)$ slopes, and a set R of $O(\Delta)$ triangles such that any nondegenerate adder A of maximum degree Δ has a straight line embedding $\mathscr{E}_A$ with the following properties:*

1. *All the edge-slopes of $\mathscr{E}_A$ belong to the set S.*
2. *All the edges on the outer face of $\mathscr{E}_A$ have slopes from the set S_0.*
3. *Each internal face of $\mathscr{E}_A$ is homothetic to a triangle from R.*
4. *The head of A coincides the edge ab of Q and the tail of A coincides with cd.*
5. *The embedding $\mathscr{E}_A$ is contained in the convex hull of $abcd$.*

Proof Note that the lemma is clearly true when restricted to adders of length at most four (or any other bounded length). In the rest of the proof, we assume that A is an adder of length at least five.

We first deal with the case when the edges ab and cd are parallel (i.e., Q is a trapezoid), and the adder A has odd length $\ell = 2k + 1$. Without loss of generality, assume that the segments ab and cd are horizontal and that the line containing cd is above the line containing ab. Let α be the slope of the diagonal ac and β the slope of the diagonal bd, with $0 < \alpha < \beta < \pi$. Let e be the point where the two diagonals intersect. Notice that the two triangles abe and cde are homothetic. Let $r = \|ab\|/\|cd\| = \|ae\|/\|ce\|$ be the dilation factor of the homothecy.

Let Z be the zigzag path of A. Let us identify the head of A with the segment ab and the tail of A with cd, in such a way that the cyclic order of the four points $abcd$ on the boundary of Q is the same as the cyclic order in which the corresponding vertices appear on the outer face of A.

Since A has odd length, the endpoints of its zigzag path are diagonally opposite in Q, see Fig. 5. We lose no generality by assuming that a and c are the endpoints of the zigzag path. Let $v_0, v_1, v_2, \dots, v_k, w_k, w_{k-1}, w_{k_2}, \dots, w_1, w_0$ be the sequence of the vertices of Z, in the order in which they appear on the path Z, with $v_0 = a$, $v_1 = b$, $w_0 = c$,

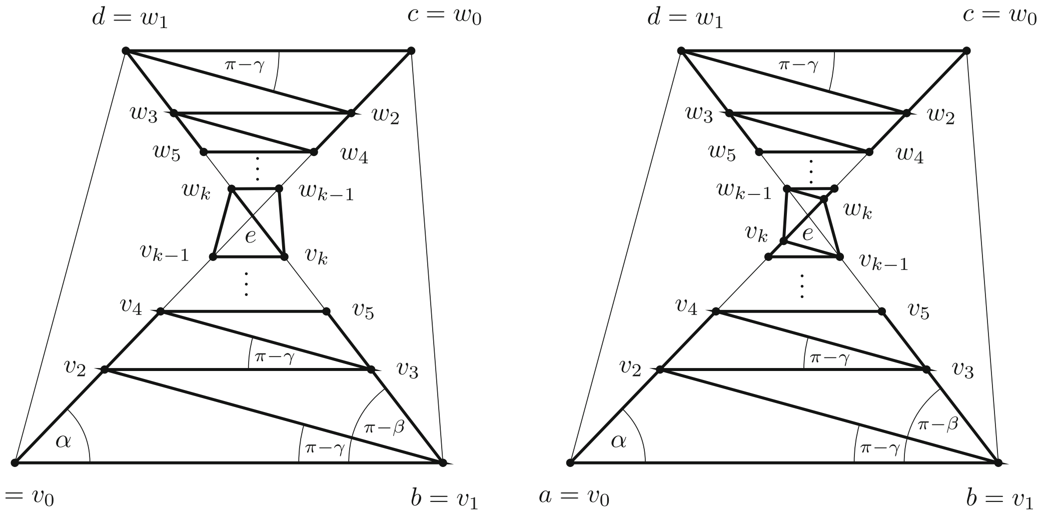


Fig. 5 Embedding an adder with prescribed head and tail. These figures illustrates the embedding of the adder of odd length $2k + 1$. The two figures correspond to the two cases depending on the parity of k

and $w_1 = d$. Fix an arbitrary slope γ such that $\beta < \gamma < \pi$. All the vertices of Z will be embedded on the two diagonals ac and bd. Since the first two and last two vertices have already been embedded, let us proceed by induction, separately in each half of Z. If, for some $i \geq 0$, the vertex v_i has already been embedded on the diagonal ac, then we embed v_{i+1} on bd in such a way that the segment $v_i v_{i+1}$ is horizontal. If v_i has been embedded on the diagonal bd, then v_{i+1} is embedded on ac and the slope of $v_i v_{i+1}$ is equal to γ.

We proceed similarly with the vertices w_i: if w_i is on ac then w_{i+1} is on bd and the segment $w_i w_{i+1}$ has slope γ; otherwise w_i is on bd and w_{i+1} is on ac and the corresponding segment is horizontal.

We may easily show by induction that for any i, the triangles $ev_i v_{i+1}$ and $ew_i w_{i+1}$ are similar, all of them with the same ratio $r = \|ev_i\|/\|ew_i\|$. Furthermore, we see that $ev_i v_{i+1}$ is similar to $ev_{i+2}v_{i+3}$, with a ratio q that is independent of i. From these facts, we see that all the segments of the form $v_i w_{i+1}$ have at most two distinct slopes (depending on the parity of i), and similarly for the segments of the form $w_i v_{i+1}$.

Let us consider all the triangles formed by triples of vertices xyz where x, y and z are three consecutive vertices of the path Z. Note that these triangles are internally disjoint, and their edges form at most six distinct slopes, namely 0, α, β, γ, the slope of the segment $v_k w_{k-1}$ and the slope of the segment $v_{k-1} w_k$. Furthermore, the latter two slopes belong to a set of at most four slopes that are independent of k, and hence independent of the adder A. The union of the above-described triangles will form the outer boundary of our embedding of A. It remains to place the vertices of A that do not belong to Z to this boundary.

Let us fix $\Delta - 2$ additional slopes $\gamma_1 < \gamma_2 < \cdots b < \gamma_{\Delta-2}$ which are all greater than γ but smaller than π. Note than any vertex u of A that does not belong to Z is incident to exactly one edge that does not belong to the outer face of A, and this edge connects u to a vertex of Z. Thus, to complete the description of the embedding of A, it suffices to specify, for every vertex v of Z, the slopes of all the edges that do not belong to the outer face of A and that connect v to a vertex not belonging to Z. Thus, let us fix an arbitrary vertex v of Z. Let us assume that v has been embedded on the diagonal ac and that $v = v_i$ for some $i \leq k$ (the cases when v belongs to bd or $v = w_i$ are analogous). Let $u_1, \ldots, u_\ell$ be the vertices not belonging to Z and adjacent to v by an internal edge of A. Note that if v has at least one such neighbor u_i, then $v \neq v_0$, because v_0 is not incident to any edge not belonging to the outer face. Let v^+ be the vertex that follows after v on Z (typically, $v^+ = v_{i+1}$, unless $v = v_k$, when $v^+ = w_k$). Assume that the vertices $u_1, \ldots u_\ell$ are listed in their counterclockwise order with respect to the neighborhood of v. Let us place each u_i at the intersection of the line $v_{i-1}v^+$ and the ray $(v, \pi + \gamma_i)$. This choice guarantees that the edge vu_i has slope γ_i.

We have thus found a straight line embedding of A that has all the required properties and uses at most $\Delta + O(1)$ slopes. This completes the case when A is an odd-length adder and Q is a trapezoid.

Assume now that A is an arbitrary nondegenerate adder of length $\ell \geq 5$, and Q is an arbitrary convex quadrilateral. Our goal is to reduce this situation to the cases solved above. Note that the adder A can be written as a union of two non-degenerate sub-adders A_1 and A_2, where A_1 has odd length, A_2 has length three or four, A_1 has the same head as A, A_2 has the same tail as A, the tail of A_1 is the head of A_2, and the adders A_1 and A_2 are otherwise disjoint. Accordingly, the convex quadrilateral $Q = abcd$ can be decomposed into a union of two internally disjoint quadrilaterals $Q_1 = abc'd'$ and $Q_2 = d'c'cd$, where Q_1 is a trapezoid. We may now use our previous arguments to construct an embedding of A_1 inside Q_1, and an embedding of A_2 inside Q_2, and combine the two embeddings into an embedding of Q satisfying the conditions of the lemma. □

We will use adders as basic building blocks in a procedure that embeds any given bubble with prescribed relevant vertices in such a way that the embedding of all the relevant vertices is chosen from a finite set of points. The following technical lemma summarizes all the key properties of the bubble embedding that we are about to construct.

Lemma 3.7 *Let $T = abc$ be an isosceles triangle with base ab, and with internal angles $\varepsilon/2, \varepsilon/2$ and $\pi - \varepsilon$. Assume that the line ab is horizontal and the point c is below the line ab. For every $\Delta > 0$ there is a set S of $O(\Delta^3)$ slopes, a set P of $O(\Delta)$ points, and a set R of $O(\Delta^3)$ triangles, such that every labelled bubble $B \in \mathscr{B}(\Delta)$ has an embedding $\mathscr{E}_B$ with the following properties.*

1. *All the edge-slopes of $\mathscr{E}_B$ belong to S.*
2. *Any relevant vertex of B is embedded at a point from P.*
3. *Every internal face of $\mathscr{E}_B$ is homothetic to a triangle from R.*
4. *The root edge of B coincides with the segment ab.*
5. *The whole embedding $\mathscr{E}_B$ is inside the triangle T.*
6. *Any relevant vertex of $\mathscr{E}_B$ has visibility in any direction from the set $\langle \pi + \varepsilon, 2\pi - \varepsilon \rangle$.*

Proof Let us first introduce some terminology (see Fig. 6). Let $B \in \mathscr{B}(\Delta)$ be a labelled bubble. Recall from Definition 3.3 that the dual of B, denoted by $\widehat{B}$, is a rooted binary tree whose root corresponds to the root face of B. For an internal face Φ of B, we let $\widehat{\Phi}$ denote the corresponding node of $\widehat{B}$. We distinguish several types

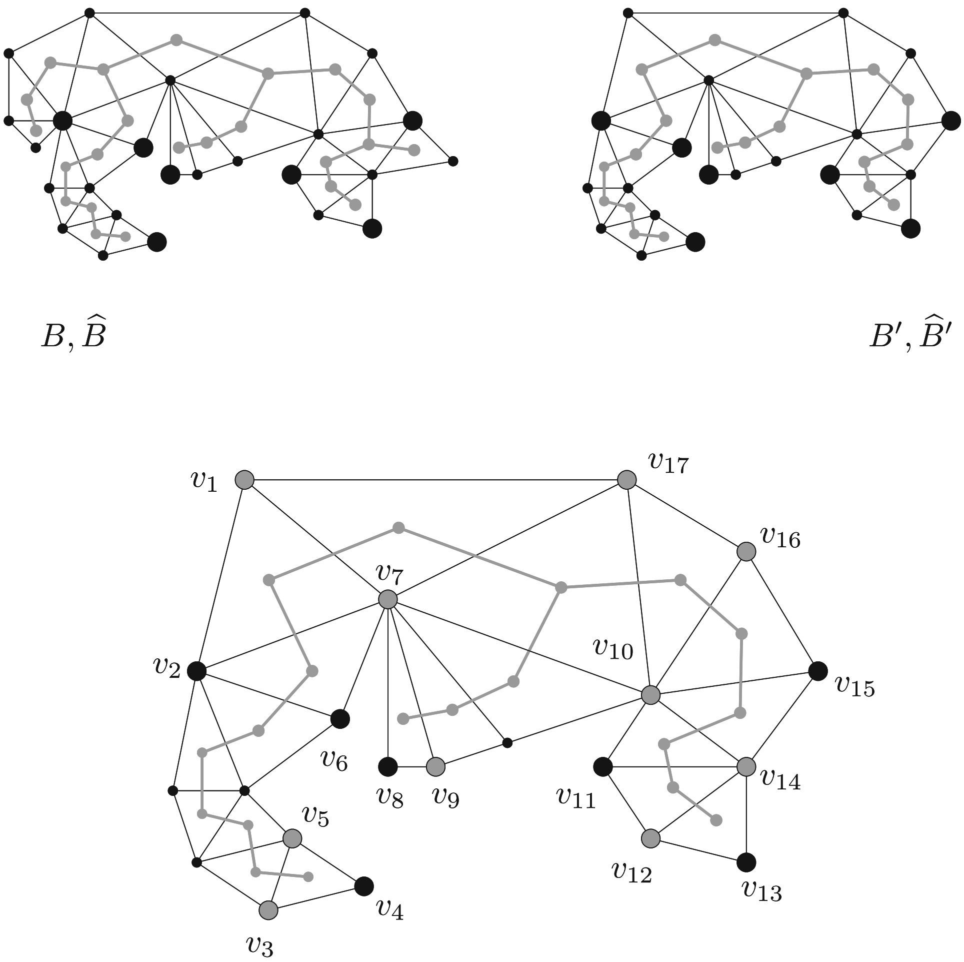


Fig. 6 An example of a labelled bubble B with its dual tree $\widehat{B}$. Relevant vertices are represented by large *black disks*. The large *gray disks* of the bottom figure represent the non-relevant priority vertices

of nodes in $\widehat{B}$. A node $\widehat{\Phi}$ is called *relevantnode*, if the bottom vertex of the face Φ is a relevant vertex of$B$. A node $\widehat{\Phi}$ of $\widehat{B}$ is called *peripheral* if the subtree of $\widehat{B}$ rooted at $\widehat{\Phi}$ does not contain any relevant node, in other words, neither $\widehat{\Phi}$ nor any descendant of $\widehat{\Phi}$ is relevant. A node is *central* if it is not peripheral. Note that the central nodes induce a subtree of $\widehat{B}$; we let $\widehat{B}'$ denote this subtree. By construction, all the leaves of $\widehat{B}'$ are relevant nodes (but there may be relevant nodes that are not leaves).

A node $\widehat{\Phi}$ of $\widehat{B}'$ is a *branching node* if both its children belong to $\widehat{B}'$ as well. A node of $\widehat{B}'$ is a *connecting node* if it is neither relevant nor branching. By definition, each connecting node has a unique child in $\widehat{B}'$, and the connecting nodes induce in $\widehat{B}'$ a disjoint union of paths. We call these paths *the connections*.

We say that a face Φ of B is a *relevant face* if the corresponding node $\widehat{\Phi}$ is a relevant node. Peripheral faces, branching faces and connecting faces are defined analogously. Let B' be the subgraph of B whose dual is $\widehat{B}'$. If $\widehat{B}'$ is empty, define B' to be the trivial bubble consisting of the root edge of B. In any case, B' is a subbubble of B and has the same root edge as B.

Note that since every leaf of $\widehat{B}'$ is a relevant node, and since B has at most 3Δ relevant vertices by definition of $\mathscr{B}(\Delta)$, the tree $\widehat{B}'$ has at most $O(\Delta)$ leaves and consequently at most $O(\Delta)$ branching nodes.

Let us now describe the basic idea of the proof. We begin by specifying the set P of points. The points of P will form a convex cup inside the triangle T. For a given bubble $B \in \mathscr{B}(\Delta)$, we construct the embedding $\mathscr{E}_B$ in three steps. In the first step, we take all the vertices of B that belong to relevant faces and branching faces, and embed them to the points of P. In the second step, we embed all the connecting faces. Each connection in $\widehat{B}'$ corresponds to a (possibly degenerate) adder contained in B', whose head and tail have been embedded in the first step. Using the construction from Lemma 3.6, we insert these adders into the embedding. Thus, in the first two steps, we construct an embedding of B'. In the third step, we extend this embedding into an embedding of B by adding the peripheral faces. These faces form a disjoint union of subbubbles, each of them rooted at an edge belonging to the outer face of B'. We use Corollary 3.1 to embed each of these subbubbles into a thin triangle above a given root edge.

Let us describe the individual steps in detail. Set $D = 18\Delta$. Recall that T is an isosceles triangle with base ab. Let C be any circular arc with endpoints a and b, drawn inside T. Choose a sequence $p_1, p_2, \ldots, p_D$ of distinct points of C, in such a way that $p_1 = a$, $p_D = b$, and the remaining points are chosen arbitrarily on C in order to form a left-to-right sequence. Let P be the set $\{p_1, \ldots, p_D\}$.

Let us say that a vertex v of B is a *priority vertex* if it either belongs to a relevant face, or it belongs to a branching face, or it belongs to the root edge of B. Note that all priority vertices actually belong to B', and that each relevant vertex is a priority vertex as well. Let ℓ be the number of priority vertices. We know that B has at most 3Δ relevant faces. Since every leaf of $\widehat{B}'$ represents a relevant face, we see that B' has at most $3\Delta - 1$ branching faces. This implies that $\ell < D = 18\Delta$.

Let $v_1, v_2, \ldots, v_\ell$ be the sequence of all the priority vertices of B, listed in counter-clockwise order of their appearance on the outer face of B, in such a way that v_1 and v_ℓ are the vertices of the root edge of B. For each $i \in \{1, \ldots, \ell - 1\}$, we embed the vertex v_i on the point p_i, while the vertex v_ℓ is embedded on the point $v_D = b$. Note that this embedding guarantees that the root edge of B coincides with the segment $ab = p_1 p_D$. Moreover, since this embedding preserves the cyclic order of the vertices on the boundary of the outer face, we know that the edges induced by the priority vertices do not cross. This completes the first step of the embedding.

In the second step, we describe the embedding of the connecting faces of B. Let $\Phi_1, \Phi_2, \ldots, \Phi_k$ be a sequence of faces of B corresponding to a connection in $\widehat{B}$, where we assume that for each $i < k$, the node $\widehat{\Phi}_i$ is the parent of $\widehat{\Phi}_{i+1}$ in $\widehat{B}$. See Fig. 7. Let x be the left vertex of Φ_1 and let y be the right vertex of Φ_1. The vertices x and y either form the root edge of B, or they belong to the parent face of Φ_1, which is either a relevant face or a branching face. In either case, both x and y are priority vertices. In particular, x corresponds to a point $p_m \in P$, and y corresponds to $p_n \in P$, for some $m < n$.

Consider now the face Φ_k. Since it is neither relevant nor branching, it has a unique child face Φ' in B'. The face Φ' is relevant or branching, so all its vertices are priority vertices. Let u be the left vertex of Φ' and let v be its right vertex. The edge uv is the intersection of Φ' and Φ_k. Let A be the adder formed by the union of the faces $\Phi_1, \ldots, \Phi_k$, with head xy and tail uv. Note that this adder does not contain any other priority vertices apart from x, y, u and v. In particular, the vertex u is either equal to x,

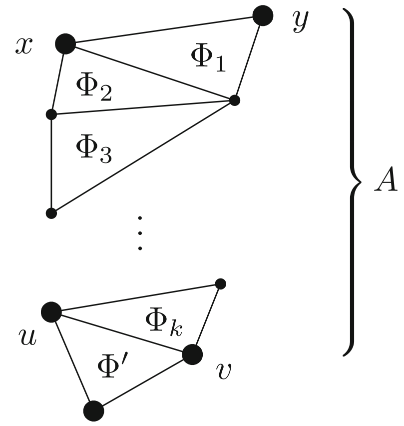


Fig. 7 An adder representing a connection in $\widehat{B}$

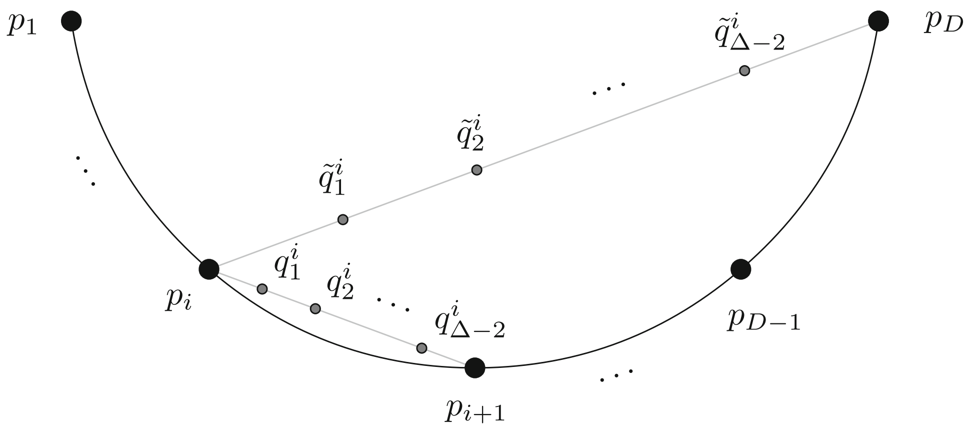


Fig. 8 The auxiliary points from the set Q

or it corresponds to p_{m+1}. For the vertex v, we have three possibilities: either $v = y$, or $v = p_{n-1}$, or $v = p_{\ell-1}$ and $y = p_D$.

Let us first deal with the case when the adder A is degenerate, i.e., either $x = u$ or $y = v$. We first define a set Q of auxiliary points (see Fig. 8). For every $i < D$, consider the segment $p_i p_{i+1}$, and subdivide this segment with $\Delta - 2$ new points $q^i_1, q^i_2, \dots, q^i_{\Delta-2}$. Next, for $i < D-1$, consider also the segment $p_i p_D$ and subdivide it with $\Delta - 2$ points $\tilde{q}^i_1, \tilde{q}^i_2, \dots, \tilde{q}^i_{\Delta-2}$. Let Q be the set of all the points q^i_j and $\tilde{q}^i_j$, for all i and j.

Assume now that A is a degenerate adder with $x = u$ (the case when $y = v$ is analogous). Recall that A has k internal faces $\Phi_1, \dots, \Phi_k$. All these faces share the vertex x, and in particular, x has degree $k + 1$ in A. This shows that $k < \Delta$, and consequently there are at most $\Delta - 2$ non-priority vertices in A, all of them on a path from v to y. See Fig. 9. If $v = p_{n-1}$, we embed these non-priority vertices on the points $q^{n-1}_1, \dots, q^{n-1}_{k-1}$. On the other hand, if $v = p_{\ell-1}$ and $y = p_D$, we embed the non-priority vertices of A on the points $\tilde{q}^{\ell-1}_1, \dots, \tilde{q}^{\ell-1}_{k-1}$. This determines the embedding of A.

Consider now the case when A is non-degenerate. The four vertices x, y, u and v form a convex quadrilateral, and we embed A inside this quadrilateral, using the construction of Lemma 3.6. This again determines the embedding of A.

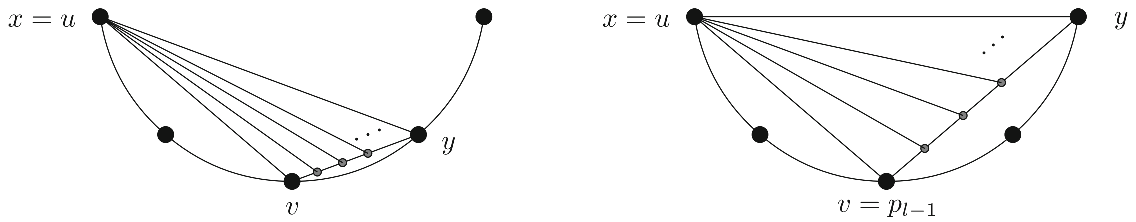


Fig. 9 The embedding of a degenerate connection adder

Using the constructions described above, we embed all the adders representing connections in $\widehat{B}$. Note that each adder is embedded inside the convex hull of its head and tail. Moreover, if A and A' are adders representing two different connections, the convex hull of the head and tail of A is disjoint from the convex hull of the head and tail of A', except for at most one vertex shared by the two adders. This shows that the embedding is indeed a plane embedding of the graph B', completing the second step of the construction.

Before we describe the last step, let us estimate the number of vertices, edge-slopes and internal faces that may arise in the first two steps. Clearly, any relevant vertex is embedded on a point from the set P, which has size $O(\Delta)$ and does not depend on the bubble B.

Any edge e embedded in the first two steps may have one of the following forms.

- The edge e connects two points from P. Such edges can take at most $O(\Delta^2)$ slopes.
- The edge e connects a vertex from P to a vertex from Q. This yields $O(\Delta^3)$ possible slopes.
- The edge e connects two vertices of Q. This is only possible when both vertices of e belong to a segment determined by a pair of points in P. The slope of e is then equal to a slope determined by two points from P.
- The edge e belongs to a non-degenerate adder A representing a connection in B. In the embedding from Lemma 3.6, the edges of a given adder A determine at most $O(\Delta)$ slopes, and these slopes only depend on the four vertices forming the head and tail of A. This fourtuple of vertices has the form $\{p_i, p_{i+1}, p_{j-1}, p_j\}$ or $\{p_i, p_{i+1}, p_{j-1}, p_D\}$. There are $O(\Delta^2)$ such fourtuples and hence $O(\Delta^3)$ possible slopes for the edges of this type.

Overall, there is a set of $O(\Delta^3)$ slopes, independent of B, such that any edge embedded in the first two steps has one of these slopes.

Next, we count homothecy types of internal faces. Any internal face Φ embedded in the first two steps has one of the following types.

- All the vertices of Φ belong to P. There are $O(\Delta^3)$ such faces.
- Φ has two vertices from P and one vertex from Q. In such case the triple of vertices of Φ must be of one of these forms, for some values of i, j and k : $\{p_i, p_j, q_1^j\}$, or $\{p_i, p_j, q_k^{j-1}\}$, or $\{p_i, p_D, \tilde{q}_k^j\}$. There are $O(\Delta^3)$ such triples.
- Φ has two vertices from Q and one vertex from P. In such case the three vertices are of the form $\{q_j^i, q_{j+1}^i, p_k\}$ or $\{\tilde{q}_j^i, \tilde{q}_{j+1}^i, p_k\}$ for some i, j and k. This again gives $O(\Delta^3)$ possibilities for Φ.

– Φ is an internal face of a non-degenerate adder, embedded by Lemma 3.6. By Lemma 3.6, the internal faces of such an adder form $O(\Delta)$ homothecy types depending only on the position of head and tail. Since there are $O(\Delta^2)$ positions for head and tail, this gives $O(\Delta^3)$ triangle types up to homothecy.

We conclude that each internal face of B' is homothetic to one of $O(\Delta^3)$ triangles, and these triangles do not depend onB'.

As we will need it later, we now estimate the number of slopes formed by edges on the outer face of B'. For e on the outer face of B' there are two possibilities.

– If both endpoints of e are priority vertices, or if e belongs to a connection represented by a degenerate adder, then the line determined by the segment e passes through two points of P. In particular, such a segment e must have one of $O(\Delta^2)$ slopes determined by P.
– Suppose e belongs to the outer face of a non-degenerate adder A. By Lemma 3.6, the edges of the outer face of A have $O(1)$ distinct slopes, depending on the head and tail of A. Overall, such edges have at most $O(\Delta^2)$ slopes.

This shows that the slopes of the edges of the outer face of B' all belong to a set of $O(\Delta^2)$ slopes.

To finish the proof, it remains to perform the third step of the construction, where we embed the peripheral faces. Fix an angle $\delta > 0$ such that $\delta < \varepsilon/2$ and any two distinct edge-slopes used in the first two steps of the construction differ by more than 2δ. Let e be an edge of the outer face of B'. Let T_e be an isosceles triangle whose base is the edge e, whose internal angles have size δ, δ, and $\pi - 2\delta$, and which lies in the outer face of B'. It is easy to check that our choice of δ guarantees that for any two edges e and f on the outer face of B', the triangles T_e and T_f are disjoint, except for a possible common vertex of e and f.

Let $\widehat{B}_0$ be a maximal subtree of $\widehat{B}$ formed entirely by peripheral nodes, and let B_0 be the dual of $\widehat{B}_0$. Note that B_0 is a subbubble of B rooted at an edge of the outer face of B'. Let e be the root edge of B_0. Using Corollary 3.1, we embed B_0 inside T_e, in such a way that the root edge of B_0 coincides with e. This embedding of B_0 uses $O(\Delta)$ edge-slopes and $O(\Delta)$ triangle types for its internal faces, and these edge-slopes and triangle types only depend on the slope of e.

Since the edges on the outer face of B' may have at most $O(\Delta^2)$ edge-slopes, we may embed all the peripheral faces of B, while using only $O(\Delta^3)$ edge-slopes and $O(\Delta^3)$ triangle types in addition to the edge-slopes and triangle types used in the first two steps of the construction.

This completes the last step of the construction. It is easy to check that in the obtained embedding of B, any relevant vertex has visibility in any direction from the set $\langle \pi + \varepsilon, 2\pi - \varepsilon \rangle$, and the remaining claims of the lemma have already been verified. $\square$

We are finally ready to give the proof of the Tripod Drawing Lemma from page 991.

Proof of Lemma 3.4 Fix a tripod $T \in \mathrm{Tr}(\Delta)$. Let X, Y, and Z be the three legs of the tripod T. The center c of the tripod will coincide with the origin of the coordinate system, and the spines of the three legs will be embedded onto three rays with slopes

0, $2\pi/3$ and $4\pi/3$ starting at the origin. We will now describe how to embed the leg X onto the horizontal ray $(c, 0)$. The embeddings of the remaining two legs are then built by an analogous procedure, rotated by $2\pi/3$ and $4\pi/3$.

Let X be a fixed leg of the tripod, represented as a sequence $D_1, D_2, \ldots, D_k$ of double bubbles, ordered from the center outwards. Recall that a bubble is called relevant if it contains at least one relevant vertex. We will also say that a double bubble is relevant if at least one of its two parts is relevant.

Define a parameter D by $D = 13\Delta$. The leg X can have at most 6Δ relevant double bubbles. A maximal consecutive sequence of the form $D_i, D_{i+1}, \ldots, D_j$ in which each element is an irrelevant double bubble will be called an *irrelevant run*. We partition X into a sequence of *parts* $P_1, P_2, \ldots, P_\ell$, where a part is either a single relevant double bubble, or a nonempty irrelevant run. Since by definition no two irrelevant runs are consecutive, we see that X has at most $12\Delta + 1 < D$ parts.

Let T_ε be an isosceles triangle with internal angles of size $\varepsilon/2, \varepsilon/2$ and $\pi - \varepsilon$ whose base edge is horizontal. From Lemma 3.7, we know that there is a set of points $P_\varepsilon \subset T_\varepsilon$ of size $O(\Delta)$, a set of slopes S_ε of size $O(\Delta^3)$ and set of triangles R_ε of size $O(\Delta^3)$ such that any bubble of $B \in \mathscr{B}(\Delta)$ can be embedded inside T_ε using slopes from S_ε in such a way that each relevant vertex of B coincides with a point from the set P_ε and the internal faces of the embedding are homothetic to triangles in R. Let $\mathscr{E}_B$ denote this embedding.

We will combine these embeddings to obtain an embedding of the whole leg X. To each of the at most D parts of X we will assign a segment of length $L = \frac{1}{D}$ on the horizontal ray $(c, 0)$.

Assume first that P_i is a part of X consisting of a single relevant double bubble, formed by a pair of bubbles B and C. We will embed P_i in such a way that the common root edge of B and C coincides with a horizontal segment e_i of length L, whose endpoints have horizontal coordinates $(i-1)L$ and iL. The two bubbles B and C are then embedded inside two scaled and translated copies of T_ε that share a common base e_i, using the embeddings $\mathscr{E}_B$ and $\mathscr{E}_C$, possibly reflected along the horizontal axis.

Now assume that P_i is a part of X that consists of an irrelevant run of k irrelevant double bubbles $D_j, D_{j+1}, \ldots, D_{i+k-1}$. We embed the root edge of each double bubble onto a segment of length L/k, and embed the rest of the double bubble into a scaled and translated copy of T_ε. We then concatenate these embeddings to obtain an embedding of the whole irrelevant run, which will occupy a segment of length exactly L on the spine of X.

Overall, since the leg has at most D parts, the whole leg will be embedded at distance at most 1 from the origin. It is easy to see that the embedding of X uses at most $2|S_\varepsilon|$ slopes and $2|R_\varepsilon|$ triangles for faces (up to scaling). The embedding of the whole tripod will then require at most $6|S_\varepsilon| = O(\Delta^3)$ slopes and $6|R_\varepsilon| = O(\Delta^3)$ non-homothetic triangles.

Let us estimate the number of possible points where a relevant vertex may be embedded. For every relevant double bubble, there are at most D possibilities where its root edge may be embedded within the embedding of X. Since a bubble may be either above or below the spine, each relevant bubble has at most $2D$ possibilities where it may appear within X, and at most $6D$ possibilities within the whole tripod. As soon as we fix the embedding of the root edge and the relative position of the bubble with respect to its spine, we are left with at most $|P_\varepsilon|$ possibilities where a relevant vertex may be embedded. There are overall at most $6D|P_\varepsilon| = O(\Delta^2)$ possible embeddings of relevant vertices.

Using Lemma 3.7, it is straightforward to check that the embedding satisfies the required visibility properties. Lemma 3.4 (and hence also Proposition 3.2 and Theorem 1.1) is now proved. □

4 Conclusion and Open Problems

We have presented an upper bound of $O(\Delta^5)$ for the planar slope number of planar partial 3-trees of maximum degree Δ. It is not obvious to us if the used methods can be generalized to a larger class of graphs, such as planar partial k-trees of bounded degree. Since a partial k-tree is a graph of tree-width at most k, it would mean generalizing our result to graphs of a larger, yet constant, tree-width.

In view of the results of Keszegh et al. [7] and Mukkamala and Szegedy [10] for the slope number of (sub)cubic planar graphs, it would also be interesting to find analogous bounds for the planar slope number.

This paper does not address lower bounds for the planar slope number in terms of Δ; this might be another direction worth pursuing.

References

1. Barát, J., Matoušek, J., Wood, D.R.: Bounded-degree graphs have arbitrarily large geometric thickness. Electron. J. Combin. **13**, R3 (2006)
2. Dujmović, V., Suderman, M., Wood, D.R.: Really straight graph drawings. In: Graph Drawings 2004. LNCS, vol. 3383, pp. 122–132 (2004)
3. Dujmović, V., Suderman, M., Wood, D.R.: Graph drawings with few slopes. Comput. Geom. **38**, 181–193 (2007)
4. Dujmović, V., Eppstein, D., Suderman, M., Wood, D.R.: Drawings of planar graphs with few slopes and segments. Comput. Geom. **38**, 194–212 (2007)
5. Jelínek, V., Jelínková, E., Kratochvíl, J., Lidický, B., Tesař, M., Vyskočil, T.: The planar slope number of planar partial 3-trees of bounded degree. In: Graph Drawing 2009. LNCS, vol. 5849, pp. 304–314 (2010)
6. Fáry, I.: On straight-line representation of planar graphs. Acta Univ. Szeged. Sect. Sci. Math. **11**, 229–233 (1948)
7. Keszegh, B., Pach, J., Pálvölgyi, D., Tóth, G.: Drawing cubic graphs with at most five slopes. In: Graph Drawing 2006. LNCS, vol. 4372, pp. 114–125 (2007)
8. Keszegh, B., Pach, J., Pálvölgyi, D.: Drawing planar graphs of bounded degree with few slopes. In: Graph Drawing 2010. LNCS, vol. 6502, pp. 293–304 (2011)
9. Kratochvíl, J., Vaner, M.: Planar and projective planar embeddings of partial 3-trees (2012, in preparation)
10. Mukkamala, P., Szegedy, M.: Geometric representation of cubic graphs with four directions. Comput. Geom. **42**, 842–851 (2009)
11. Mukkamala, P., Pálvölgyi, D.: Drawing cubic graphs with the four basic slopes. In: Graph Drawing 2011. LNCS, vol. 7034, pp. 254–265 (2012)
12. Pach, J., Pálvölgyi, D.: Bounded-degree graphs can have arbitrarily large slope numbers. Electr. J. Combin. **13**, N1 (2006)
13. Wade, G.A., Chu, J.-H.: Drawability of complete graphs using a minimal slope set. Comput. J. **37**, 139–142 (1994)

Appendix F

Kuratowski-Type Theorem for Planarity of PEGs

Computational Geometry 46 (2013) 466–492

Contents lists available at SciVerse ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

A Kuratowski-type theorem for planarity of partially embedded graphs ☆

Vít Jelínek [a], Jan Kratochvíl [b], Ignaz Rutter [c,*]

[a] Computer Science Institute, Charles University, Malostranské nám. 25, 118 00 Praha 1, Czech Republic
[b] Department of Applied Mathematics and Institute for Theoretical Computer Science, Charles University, Malostranské nám. 25, 118 00 Praha 1, Czech Republic
[c] Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Box 6980, D-76128 Karlsruhe, Germany

ARTICLE INFO

Article history:
Received 3 July 2011
Received in revised form 13 April 2012
Accepted 25 July 2012
Available online 2 November 2012
Communicated by F. Hurtado and M. van Kreveld

Keywords:
Planar graphs
Partially embedded graphs
Kuratowski theorem

ABSTRACT

A partially embedded graph (or Peg) is a triple $(G, H, \mathcal{H})$, where G is a graph, H is a subgraph of G, and $\mathcal{H}$ is a planar embedding of H. We say that a Peg $(G, H, \mathcal{H})$ is planar if the graph G has a planar embedding that extends the embedding $\mathcal{H}$.
We introduce a containment relation of Pegs analogous to graph minor containment, and characterize the minimal non-planar Pegs with respect to this relation. We show that all the minimal non-planar Pegs except for finitely many belong to a single easily recognizable and explicitly described infinite family. We also describe a more complicated containment relation which only has a finite number of minimal non-planar Pegs.
Furthermore, by extending an existing planarity test for Pegs, we obtain a polynomial-time algorithm which, for a given Peg, either produces a planar embedding or identifies an obstruction.

1. Introduction

A *partially embedded graph* (Peg) is a triple $(G, H, \mathcal{H})$, where G is a graph, H is a subgraph of G, and $\mathcal{H}$ is a planar embedding of H. The problem PartiallyEmbeddedPlanarity (Pep) asks whether a Peg $(G, H, \mathcal{H})$ admits a planar (non-crossing) embedding of G whose restriction to H is $\mathcal{H}$. In this case we say that the Peg $(G, H, \mathcal{H})$ is *planar*. Despite of this being a very natural generalization of planarity, this approach has been considered only recently [1]. It should be mentioned that all previous planarity testing algorithms have been of little use for Pep, as they all allow flipping of already drawn parts of the graph, and thus are not suitable for preserving an embedding of a given subgraph.

It is shown in [1] that planarity of Pegs can be tested in linear time. In this paper we complement the algorithm in [1] by a study of the combinatorial aspects of this question. In particular, we provide a complete characterization of planar Pegs via a small set of forbidden substructures, similarly to the celebrated Kuratowski theorem [12], which characterizes planarity via the forbidden subdivisions of K_5 and $K_{3,3}$, and the closely related theorem of Wagner [14], which characterizes planarity via forbidden K_5 and $K_{3,3}$ minors. Our characterization can then be used to modify the existing planarity test for partially embedded graphs into a certifying algorithm that either finds a solution or finds a certificate, i.e., a forbidden substructure, that shows that the instance is not planar.

Understanding the forbidden substructures may be particularly beneficial in studying the problem *simultaneous embedding with fixed edges*, or SEFE for short, which asks whether two graphs G_1 and G_2 on the same vertex set V admit two drawings Γ_1 and Γ_2 of G_1 and G_2, respectively, such that (i) all vertices are mapped to the same point in Γ_1 and Γ_2, (ii) each drawing Γ_i is a planar drawing of G_i for $i = 1, 2$, and (iii) edges common to G_1 and G_2 are represented by the same Jordan curve

☆ Supported by the GraDR–EuroGIGA project No. GIG/11/E023.
* Corresponding author.
E-mail addresses: jelinek@iuuk.mff.cuni.cz (V. Jelínek), honza@kam.mff.cuni.cz (J. Kratochvíl), rutter@kit.edu (I. Rutter).

0925-7721/$ – see front matter
http://dx.doi.org/10.1016/j.comgeo.2012.07.005

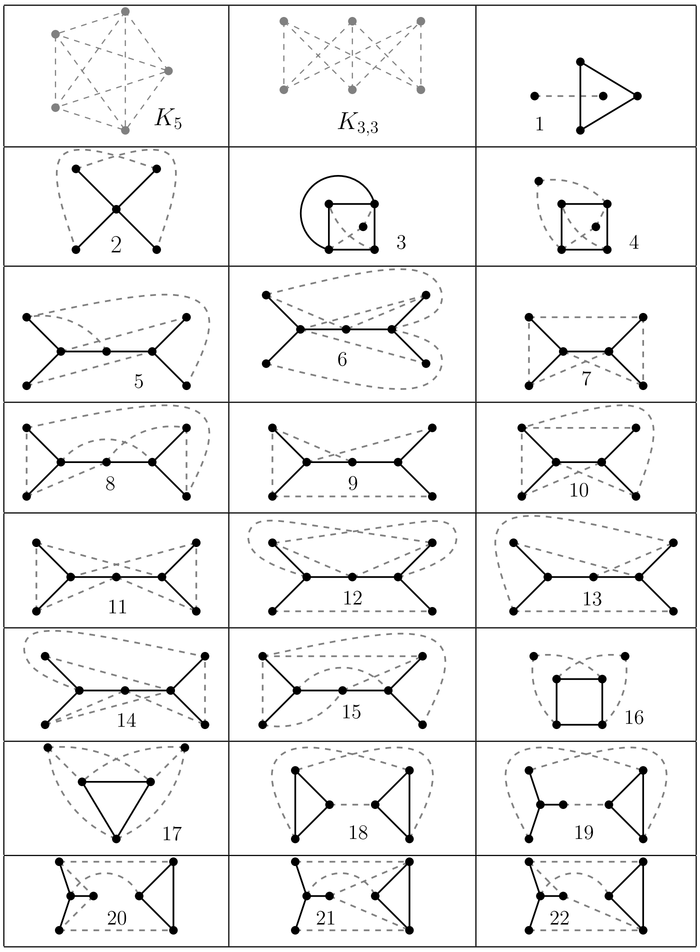


Fig. 1. The obstructions not equal to the k-fold alternating chains for $k \geqslant 4$. The black solid edges belong to H, the light dashed edges to G, but not to H. All the vertices belong to both G and H, except for K_5 and $K_{3,3}$, where H is empty.

in Γ_1 and Γ_2. Jünger and Schulz [8] show that two graphs admit a SEFE if and only if they admit planar embeddings that coincide on the intersection graph. In this sense, our obstructions give an understanding of which configurations should be avoided when looking for an embedding of the intersection graph.

For the purposes of our characterization, we introduce a set of operations, called PEG-*minor operations*, that preserve the planarity of PEGS. Note that it is not possible to use the usual minor operations, as sometimes, when contracting an edge of G not belonging to H, it is not clear how to modify the embedding of H. Our minor-like operations are defined in Section 2.

Our goal is to identify all minimal non-planar PEGS in the minor-like order determined by our operations; such PEGS are referred to as *obstructions*. Our main theorem says that all obstructions are depicted in Fig. 1 or belong to a well described infinite class of so-called *alternating chains* (the somewhat technical definition is postponed to Section 2). It can be verified that each of them is indeed an obstruction, i.e., it is not planar, but applying any of the PEG-minor operations results in a planar PEG.

We say that a PEG *avoids* a PEG X if it does not contain X as a PEG-minor. Furthermore, we say that a PEG is *obstruction-free* if it avoids all PEGS of Fig. 1 and all alternating chains of lengths $k \geqslant 4$. Then our main theorem can be expressed as follows.

Theorem 1. *A* PEG *is planar if and only if it is obstruction-free.*

Since our PEG-minor operations preserve planarity, and since all the listed obstructions are non-planar, any planar PEG is obstruction-free. The main task is to prove that an obstruction-free PEG is planar.

Having identified the obstructions, a natural question is if the PEG-planarity testing algorithm of [1] can be extended so that it provides an obstruction if the input is non-planar. It is indeed so.

Theorem 2. *There is a polynomial-time algorithm that for an input* PEG $(G, H, \mathcal{H})$ *either constructs a planar embedding of G extending $\mathcal{H}$, or provides a certificate of non-planarity, i.e., identifies an obstruction present in $(G, H, \mathcal{H})$ as a* PEG*-minor.*

The paper is organized as follows. In Section 2, we first recall some basic definitions and results on PEGS and their planarity, and then define the PEG-minor order and the alternating chain obstructions. In Section 3, we show that the main theorem holds for instances where G is biconnected. We extend the main theorem to general (not necessarily biconnected) PEGS in Section 4. In Section 5, we present possible strengthening of our PEG-minor relations, and show that when more complicated reduction rules are allowed, the modified PEG-minor order has only finitely many non-planar PEGS. In Section 6 we briefly provide an argument for Theorem 2 and then conclude with some open problems.

2. Preliminaries and notation

2.1. Embeddings

A *drawing* of a graph is a mapping of each vertex to a distinct point in the plane and of each edge to a simple Jordan curve that connects its endpoints. A drawing is *planar* if the curves representing the edges intersect only in common endpoints. A graph is *planar* if it admits a planar drawing. Such a planar drawing determines a subdivision of the plane into connected regions, called *faces*, and a circular ordering of the edges incident to each vertex, called *rotation scheme*. Traversing the border of a face F in such a way that the face is to the left yields a set of circular lists of vertices, the *boundary* of F. Note that the boundary of a face is not necessarily connected if the graph is not connected and that vertices can be visited several times if the graph is not biconnected. The boundary of a face F can be uniquely decomposed into a set of simple edge-disjoint cycles, bridges (i.e., edges that are not part of a cycle) and isolated vertices. We orient these cycles so that F is to their left to obtain the *facial cycles* of F.

Two drawings are *topologically equivalent* if they have the same rotation scheme and, for each facial cycle, the vertices to its left are the same in both drawings. A *planar embedding* is an equivalence class of planar drawings. Let $\mathcal{G}$ be a planar embedding of G and let H be a subgraph of G. The *restriction* of $\mathcal{G}$ to H is the embedding of H that is obtained from $\mathcal{G}$ by considering only the vertices and the edges of H. We say that $\mathcal{G}$ is an *extension* of a planar embedding $\mathcal{H}$ of H if the restriction of $\mathcal{G}$ to H is $\mathcal{H}$.

2.2. Connectivity and SPQR-trees

A graph is *connected* if any pair of its vertices is connected by a path. A maximal connected subgraph of a graph G is a *connected component* of G. A *cut-vertex* is a vertex $x \in V(G)$ such that $G - x$ has more components than G. A connected graph with at least three vertices is *2-connected* (or *biconnected*) if it has no cut-vertex. In a biconnected graph G, a *separating pair* is a pair of vertices $\{x, y\}$ such that $G - x - y$ has more components than G. A biconnected graph with at least four vertices is *3-connected* if it has no separating pair. We say that a PEG $(G, H, \mathcal{H})$ is *connected*, *biconnected* and *3-connected* if G is connected, biconnected and 3-connected, respectively. An edge of a graph G is sometimes referred to as a G*-edge*, and a path in G is a G*-path*.

A connected graph can be decomposed into its maximal biconnected subgraphs, called *blocks*. Each edge of a graph belongs to exactly one block, only cut-vertices are shared between different blocks. This gives rise to the block-cutvertex tree of a connected graph G, whose nodes are the blocks and cut-vertices of G, and whose edges connect cut-vertices to blocks they belong to.

The planar embeddings of a 2-connected graph can be represented by the SPQR-tree, which is a data structure introduced by Di Battista and Tamassia [3,4]. A more detailed description of the SPQR-tree can be found in the literature [3–5,13]. Here we just give a sketch and some notation.

The SPQR-tree $\mathcal{T}$ of a 2-connected graph G is an unrooted tree that has four different types of nodes, namely S-, P-, Q- and R-nodes. The Q-nodes are the leaves of $\mathcal{T}$, and they correspond to edges of G. Each internal node μ of $\mathcal{T}$ has an associated biconnected multigraph $\mathfrak{S}$, its *skeleton*, which can be seen as a simplified version of the graph G. The subtrees of μ in $\mathcal{T}$ when $\mathcal{T}$ is rooted at μ determine a decomposition of G into edge-disjoint subgraphs $G_1, \ldots, G_k$, each of which is connected and shares exactly two vertices u_i, v_i with the rest of the graph G. Each G_i is represented in the skeleton of μ by an edge e_i connecting u_i and v_i. We say that G_i is the *pertinent graph* of the edge e_i. We also say that the skeleton edge e_i *contains* a vertex v or an edge e of G, or that v and e *project* into e_i, if v or e belong to the pertinent graph G_i of e_i. For a subgraph G' of G, we say that G' *intersects* a skeleton edge e_i, if at least one edge of G' belongs to G_i.

The skeleton of an S-node is a cycle of length $k \geqslant 3$, the skeleton of a P-node consists of $k \geqslant 3$ parallel edges, and the skeleton of an R-node is a 3-connected planar graph. The SPQR-tree of a planar 2-connected graph G represents all planar embeddings of G in the sense that choosing planar embeddings for all skeletons of $\mathcal{T}$ corresponds to choosing a planar embedding of G and vice versa.

Suppose that $e = uv$ is an edge of the skeleton of a node μ of an SPQR-tree of a biconnected graph G, and let G_e be the pertinent graph of e. The graph G_e satisfies some additional restrictions depending on the type of μ: if μ is an S-node, then G_e is biconnected, and if μ is a P-node, then either G_e is a single edge uv or $G_e - \{u, v\}$ is connected. Regardless of the type of μ, every cut-vertex in G_e separates u from v, otherwise G would not be biconnected.

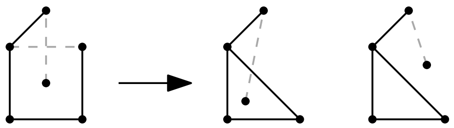


Fig. 2. An example of a planar Peg (left) in which a contraction of a G-edge may result in two distinct Pegs, one of which is non-planar.

2.3. Peg-*minor operations*

We first introduce a set of operations that preserve planarity when applied to a Peg $I = (G, H, \mathcal{H})$. The set of operations is chosen so that the resulting instance $I' = (G', H', \mathcal{H}')$ is again a Peg (i.e., H' is a subgraph of G' and $\mathcal{H}'$ is a planar embedding of H'). It is not possible to use the usual minor operations, as sometimes, when contracting an edge of $G - H$, the embedding of the modified graph H is not unique and some of the possible embeddings lead to planar Pegs, while some do not. This happens, e.g., when a contraction of a G-edge creates a new cycle of H-edges, in which case it is not clear on which side of this cycle the remaining components of H should be embedded (see Fig. 2).

We will consider seven minor-like operations, of which the first five are straightforward.

1. Vertex removal: Remove from G and H a vertex $v \in V(G)$ with all its incident edges.
2. Edge removal: Remove from G and H an edge $e \in E(G)$.
3. Vertex relaxation: For a vertex $v \in H$ remove v and all its incident edges from H, but keep them in G. In other words, vertex v no longer has a prescribed embedding.
4. Edge relaxation: Remove an edge $e \in E(H)$ from H, but keep it in G.
5. H-edge contraction: Contract an edge $e \in E(H)$ in both G and H, update $\mathcal{H}$ accordingly.

The contraction of G-edges is tricky, as we have to care about two things. First, we have to take care that the modified subgraph H' remains planar and second, even if it remains planar, we do not want to create a new cycle C in H as in this case the relative positions of the connected components of H with respect to this cycle may not be uniquely determined. We therefore have special requirements for the G-edges that may be contracted and we distinguish two types, one of which trivially ensures the above two conditions and one that explicitly ensures them.

6. Simple G-edge contraction: Assume that $e = uv$ is an edge of G, such that at least one of the two vertices u and v does not belong to H. Contract e in G, and leave H and $\mathcal{H}$ unchanged.
7. Complicated G-edge contraction: Assume that $e = uv$ is an edge of G, such that u and v belong to distinct components of H, but share a common face of $\mathcal{H}$. Assume further that both u and v have degree at most 1 in H. This implies that we may uniquely extend $\mathcal{H}$ to an embedding $\mathcal{H}^+$ of the graph H^+ that is obtained from H by adding the edge uv. Afterwards we perform an H-edge contraction of the edge uv to obtain the new Peg.

If a contraction produces multiple edges, we only preserve a single edge from each such set of multiple edges, so that G and H remain simple. Note that the resulting embedding $\mathcal{H}$ may depend on which edge we decide to preserve.

Let $(G, H, \mathcal{H})$ be a Peg and let $(G', H', \mathcal{H}')$ be the result of one of the above operations on $(G, H, \mathcal{H})$. The extra conditions on G-edge contractions ensure that the embedding $\mathcal{H}'$ is uniquely determined from the embedding of H after such contraction. The conditions on vertex degrees in H ensure that the rotation scheme of the H'-edges around the vertex created by the contraction is unique. In the complicated G-edge contraction, the requirement that the endpoints need to lie in distinct connected components of H that share a face ensures that the contraction does not create a new cycle in H' and that H' has a unique planar embedding induced by $\mathcal{H}$.

It is not hard to see that an embedding $\mathcal{G}$ of G that extends $\mathcal{H}$ can be transformed into an embedding $\mathcal{G}'$ of G' that extends $\mathcal{H}'$. Therefore, all the above operations preserve planarity of Pegs. If a Peg A can be obtained from a Peg B by applying a sequence of the above operations, we say that A is a Peg-*minor* of B or that B *contains* A *as a* Peg-*minor*.

2.4. *Alternating chains*

Apart from the obstructions in Fig. 1, there is an infinite family of obstructions, which we call *the alternating chains*. To describe them, we need some terminology. Let C be a cycle of length at least four, and let u, v, x and y be four distinct vertices of C. We say that the pair of vertices $\{u, v\}$ *alternates* with the pair $\{x, y\}$ on C, if u and v belong to distinct components of $C - x - y$.

Intuitively, an alternating chain consists of a cycle C of H and a sequence of internally disjoint paths $P_1, \dots, P_k$ of which only the endpoints belong to C, such that for each $i = 1, \dots, k-1$, the endpoints of P_i alternate with the endpoints of P_{i+1} on C, and no other pair of paths has alternating endpoints. Now assume that P_1 contains a vertex that is prescribed inside C. Due to the fact that the endpoints of consecutive paths alternate this implies that all P_i with i odd must be embedded inside C, while all P_i with i even must be embedded outside. A k-fold alternating chain is such that the last path P_k is prescribed in a way that contradicts this, i.e., it is prescribed inside C if k is even and outside, if k is odd.

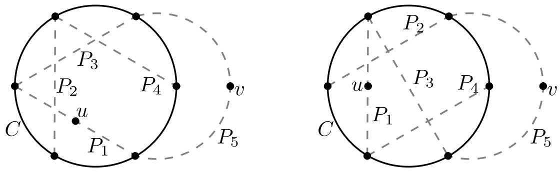


Fig. 3. Two non-isomorphic 5-fold alternating chains.

Generally it is sufficient to have paths of length 1 for $P_2, \dots, P_{k-1}$ and to have a single vertex (for the prescription) in each of P_1 and P_k. We now give a precise definition.

Let $k \geqslant 3$ be an integer. A *k-fold alternating chain* is a PEG $(G, H, \mathcal{H})$ of the following form:

- The graph H consists of a cycle C of length $k+1$ and two isolated vertices u and v. If k is odd, then u and v are embedded on opposite sides of C in $\mathcal{H}$, otherwise they are embedded on the same side.
- The graph G has the same vertex set as H, and the edges of G that do not belong to H form k edge-disjoint paths $P_1, \dots, P_k$, whose endpoints belong to C. The path P_1 has two edges and contains u as its middle vertex, the path P_k has two edges and contains v as its middle vertex, and all the other paths have only one edge.
- The endpoints of the path P_i alternate with the endpoints of the path P_j on C if and only if $j = i+1$ or $i = j+1$.
- All the vertices of C have degree 4 in G (i.e., each of them is a common endpoint of two of the paths P_i), with the exception of two vertices of C that have degree three. One of these two vertices is an endpoint of P_2, and the other is an endpoint of P_{k-1}.

Let Ach_k denote the set of k-fold alternating chains. It can be checked that for each $k \geqslant 4$, the elements of Ach_k are obstructions; see Lemma 19. Obstruction 4 from Fig. 1 is actually the unique member of Ach_3, and is an obstruction as well. However, we prefer to present it separately as an 'exceptional' obstruction, because we often need to refer to it explicitly. Note that for $k \geqslant 5$ we may have more than one non-isomorphic k-fold chain; see Fig. 3.

3. Biconnected pegs

In this section we prove Theorem 1 for biconnected PEGS. We first recall a characterization of biconnected planar PEGS via SPQR-trees.

Definition 3. Let $(G, H, \mathcal{H})$ be a biconnected PEG.

A planar embedding of the skeleton of a node of the SPQR-tree of G is *edge-compatible with* $\mathcal{H}$ if, for every vertex x of the skeleton and for every three edges of H incident to x that project to different edges of the skeleton, their order determined by the embedding of the skeleton is the same as their order around x in $\mathcal{H}$.

A planar embedding of the skeleton $\mathfrak{S}$ of a node μ of the SPQR-tree of G is *cycle-compatible with* $\mathcal{H}$ if, for every facial cycle $\vec{C}$ of $\mathcal{H}$ whose edges project to a simple cycle $\vec{C}'$ in $\mathfrak{S}$, all the vertices of $\mathfrak{S}$ that lie to the left of $\vec{C}$ and all the skeleton edges not belonging to $\vec{C}$ that contain vertices that lie to the left of $\vec{C}$ in $\mathcal{H}$ are embedded to the left of $\vec{C}'$; and analogously for the vertices to the right of $\vec{C}$.

A planar embedding of a skeleton of a node of the SPQR-tree of G is *compatible* if it is both edge- and cycle-compatible.

Angelini et al. showed that a biconnected PEG is planar if and only if the skeleton of each node admits a compatible embedding [1, Theorem 3.1]. We use this characterization and show that any skeleton of a biconnected PEG that avoids all obstructions admits a compatible embedding. Since skeletons of S-nodes have only one embedding, and their embedding is always compatible, we consider P- and R-nodes only. The two types of nodes are handled separately in Sections 3.1 and 3.2, respectively.

The following lemma will be useful in several parts of the proof.

Lemma 4. *Let $(G, H, \mathcal{H})$ be a PEG, let u be a vertex of a skeleton $\mathfrak{S}$ of a node μ of the SPQR-tree of G, and let e be an edge of $\mathfrak{S}$ with endpoints u and v. Let $F \subseteq E(H)$ be the set of edges of H that are incident to u and project into e. If the edges of F do not form an interval in the rotation scheme of u in $\mathcal{H}$ then $(G, H, \mathcal{H})$ contains obstruction 2.*

Proof. If F is not an interval in the rotation scheme, then there exist edges $f, f' \in F$ and $g, g' \in E(H) \setminus F$, all incident to u, and appearing in the cyclic order f, g, f', g' around u in $\mathcal{H}$. Let x and x' be the endpoints of f and f' different from u and let y and y' be the endpoints of g and g' different from u. For any skeleton edge f, we denote with G_f the pertinent graph of f.

If μ is an S-node, then g and g' project to the same skeleton edge uw with $v \neq w$. Note that G_{uv} and G_{uw} share only the vertex u and moreover, they are both connected even after removing u. Therefore, there exist disjoint paths P in G_{uv}

and Q in G_{uw} connecting x to x' and y to y', respectively. We may relax all internal vertices and all edges of P and Q, and then perform simple edge contractions to replace each of the two paths with a single edge. This yields obstruction 2.

If μ is an R-node, then $G_{uv} - u$ is connected, and hence it contains a path P from x to x'. Moreover, since $G - G_{uv}$ is connected, it has a path Q from y to y'. As in the previous case, contraction of P and Q yields obstruction 2.

If μ is a P-node, then $G_e - \{u, v\}$ is connected, and therefore there is a path P connecting x to x' in $G_e - \{u, v\}$. Analogous to the previous cases, a path Q from y to y' exists that avoids u and P. Again their contraction yields obstruction 2. □

In the following, we assume that the H-edges around each vertex of a skeleton that project to the same skeleton edge form an interval in the rotation scheme of this vertex.

3.1. P-nodes

Throughout this section, we assume that $(G, H, \mathcal{H})$ is a biconnected obstruction-free Peg. We fix a P-node μ of the SPQR-tree of G, and we let $\mathfrak{P}$ be its skeleton. Let u and v be the two vertices of $\mathfrak{P}$, and let $e_1, \ldots, e_k$ be its edges. Let G_i be the pertinent graph of e_i. Recall that G_i is either a single edge connecting u and v, or it does not contain the edge uv and $G_i - \{u, v\}$ is connected.

The goal of this section is to prove that $\mathfrak{P}$ admits a compatible embedding. We first deal with edge-compatibility.

Lemma 5. *Let $(G, H, \mathcal{H})$ be a biconnected obstruction-free* Peg. *Then every P-skeleton $\mathfrak{P}$ has an edge-compatible embedding.*

Proof. If $\mathfrak{P}$ has no edge-compatible embedding, then the rotation scheme around u conflicts with the rotation scheme around v. This implies that there is a triplet of skeleton edges e_a, e_b, e_c, for which the rotation scheme around u imposes a different cyclic order than the rotation scheme around v. We distinguish two cases.

Case 1. The graph H has a cycle C whose edges intersect two of the three skeleton edges, say e_a and e_b. Then the edge e_c must contain a vertex x whose prescribed embedding is to the left of C, as well as a vertex y whose prescribed embedding is to the right of C. Since x and y are connected by a path in $G_c - \{u, v\}$, we obtain obstruction 1.

Case 2. The graph H has no cycle that intersects two of the three $\mathfrak{P}$-edges e_a, e_b, e_c. Each of the three $\mathfrak{P}$-edges contains an edge of H adjacent to u as well as an edge of H adjacent to v. Since $G_i - \{u, v\}$ is connected for each i, it follows that each of the three skeleton edges contains a path from u to v, such that the first and the last edge of the path belong to H. Fix such paths P_a, P_b and P_c, projecting into e_a, e_b and e_c, respectively.

At least two of these paths (P_a and P_b, say) also contain an edge not belonging to H, otherwise they would form a cycle of H intersecting two skeleton edges. By relaxations and simple contractions, we may reduce P_a to a path of length three, whose first and last edge belong to $E(H)$ and the middle edge belongs to $E(G) \setminus E(H)$. The same reduction can be performed with P_b. The path P_c can then be contracted to a single vertex, to obtain obstruction 2. □

Next, we consider cycle-compatibility. Assume that H has at least one facial cycle whose edges intersect two distinct skeleton edges. It follows that u and v belong to the same connected component of H; denote this component by H_{uv}. We call a *uv-cycle* any facial cycle of H that contains both u and v. Note that any uv-cycle is also a facial cycle of H_{uv}, and a facial cycle of H_{uv} that contains both u and v is a uv-cycle. Following the conventions of [1], we assume that all facial cycles are oriented in such a way that a face is to the left of its facial cycles. The next lemma shows that the vertices of H_{uv} cannot violate any cycle-compatibility constraints without violating edge-compatibility as well.

Lemma 6. *Assume that C is a uv-cycle that intersects two distinct $\mathfrak{P}$-edges e_a and e_b, and that x is a vertex of H_{uv} not belonging to C. In any edge-compatible embedding of $\mathfrak{P}$, the vertex x does not violate cycle-compatibility with respect to C.*

Proof. The vertex x belongs to a skeleton edge e_x different from e_a and e_b, otherwise it cannot violate cycle-compatibility. Note that since x is in H_{uv}, e_x must contain a path P of H that connects x to one of the poles u and v. In the graph H, all the vertices of P must be embedded on the same side of C as the vertex x. The last edge of P may not violate edge-compatibility, which forces the whole edge e_x, and thus x, to be embedded on the correct side of the projection of C, as claimed. □

The next lemma shows that for an obstruction-free Peg, all vertices of H projecting to the same $\mathfrak{P}$-edge impose the same cycle-compatibility constraints for the placement of this edge.

Lemma 7. *Let x and y be two vertices of H, both distinct from u and v. Suppose that x and y project to the same $\mathfrak{P}$-edge e_a. Let C be a cycle of H that is edge-disjoint from G_a. Then x and y are embedded on the same side of C in $\mathcal{H}$.*

Proof. Since $G_a - \{u, v\}$ is a connected subgraph of G, there is a path P in G that connects x to y and avoids u and v. Since C is edge-disjoint from G_a, the path P avoids all the vertices of C. If x and y were not embedded on the same side of C in $\mathcal{H}$, we would obtain obstruction 1 by contracting C and P. □

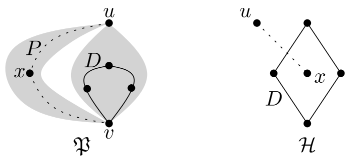


Fig. 4. Illustration of Case 1 in the proof of Proposition 8. The shaded regions represent the edges of $\mathfrak{P}$.

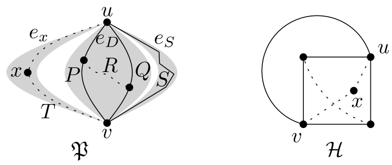


Fig. 5. Illustration of Case 2.a in the proof of Proposition 8. The shaded regions represent the edges of $\mathfrak{P}$.

We now prove the main result of this subsection.

Proposition 8. *Let $(G, H, \mathcal{H})$ be a biconnected obstruction-free* Peg*. Then every P-skeleton $\mathfrak{P}$ of the SPQR-tree of G admits a compatible embedding.*

Proof. Fix an edge-compatible embedding that minimizes the number of violated cycle-compatibility constraints; more precisely, fix an embedding of $\mathfrak{P}$ that minimizes the number of pairs (C, x) where C is a facial cycle of $\mathcal{H}$ projecting to a cycle C' of $\mathfrak{P}$, x is a vertex of $H - \{u, v\}$ projecting into a skeleton edge e_x not belonging to C', and the relative position of C' and e_x in the embedding of $\mathfrak{P}$ is different from the relative position of C and x in $\mathcal{H}$. We claim that the chosen embedding of $\mathfrak{P}$ is compatible.

For contradiction, assume that there is at least one pair (C, x) that violates cycle-compatibility in the sense described above. Let e_x be the $\mathfrak{P}$-edge containing x. Note that e_x does not contain any edge of H adjacent to u or v. If it contained such an edge, it would contain a vertex y from the component H_{uv}, and this would contradict Lemma 6 or Lemma 7. Thus, the edge e_x does not participate in any edge-compatibility constraints.

It follows that x does not belong to the component H_{uv}. That means that in $\mathcal{H}$, the vertex x is embedded in the interior of a unique face F of H_{uv}. We distinguish two cases, depending on whether the boundary of F contains both poles u and v of $\mathfrak{P}$ or not.

Case 1. The boundary of F contains at most one of the two poles u and v; see Fig. 4. Without loss of generality, the boundary of F does not contain u. Thus, F has a facial cycle D that separates u from x. The pertinent graph G_x of e_x contains a path P from x to u that avoids v. The path P does not contain any vertex of H_{uv} except u, and in particular, it does not contain any vertex of D. Contracting D to a triangle and P to an edge yields obstruction 1, which is a contradiction.

Case 2. The boundary of F contains both poles u and v of the skeleton. In this case, since u and v belong to the same block of H, the face F has a unique facial cycle D that contains both u and v. The cycle D is the only uv-cycle that has x to its left (i.e., inside its corresponding face).

The cycle D may be expressed as a union of two paths P and Q connecting u and v, where P is directed from u to v and Q is directed from v to u. We distinguish two subcases, depending on whether the paths P and Q project to different $\mathfrak{P}$-edges.

Case 2.a. Both P and Q project to the same skeleton edge e_D. This case is depicted in Fig. 5. Each of the two paths P and Q has at least one internal vertex. Since all these internal vertices are inside a single skeleton edge, there must be a path R in G connecting an internal vertex of P to an internal vertex of Q and avoiding both u and v. By choosing R as short as possible, we may assume that no internal vertex of R belongs to D. Furthermore, since $\mathfrak{P}$ by hypothesis has at least one violated cycle-compatibility constraint, it must contain at least two edges that contain an H-path from u to v. In particular, there must exist a $\mathfrak{P}$-edge e_S different from e_D that contains an H-path S from u to v.

Necessarily, the path S is embedded outside the face F, i.e., the right of D. And finally, the edge e_x must contain a G-path T from u to v that contains x. Note that e_x is different from e_D and e_S, because e_x has no H-edge incident to u or v. Thus, the paths P, Q, S, T are all internally disjoint. The five paths P, Q, R, S, and T can then be contracted to form obstruction 3.

Case 2.b. The two paths P and Q belong to distinct skeleton edges e_P and e_Q. That means that the facial cycle D projects to a cycle D' of the skeleton, formed by the two edges. Modify the embedding of the skeleton by moving e_x so that it is to the left of D'. This change does not violate edge-compatibility, because e_x has no H-edge adjacent to u or v.

We claim that in the new skeleton embedding, x does not participate in any violated cycle-compatibility constraint. To see this, we need to check that x is embedded to the right of any facial cycle $B \neq D$ of H_{uv} that projects to a cycle in the skeleton. Choose such a cycle B and let B' be its projection; see Fig. 6. Let e^+ or e^- denote the edges of D incident to u with e^+ being oriented towards u and e^- out of u. Similarly, let f^+ and f^- be the incoming and outgoing edges of B

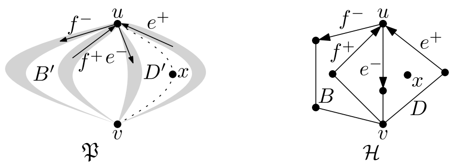


Fig. 6. Illustration of Case 2.b in the proof of Proposition 8. The left part represents the embedding of $\mathfrak{P}$ after e_x has been moved to the left of D'. Edge-compatibility guarantees that x is now on the correct side of every facial cycle in $\mathfrak{P}$.

adjacent to u. In $\mathcal{H}$, the four edges must visit u in the clockwise order (e^+, e^-, f^+, f^-), with the possibility that $e^- = f^+$ and $e^+ = f^-$.

Since the embedding of the skeleton is edge-compatible, this means that any skeleton edge embedded to the left of D' is also to the right of B', as needed. We conclude that in the new embedding of $\mathfrak{P}$, the vertex x does not violate any cycle-compatibility constraint, and by Lemma 7, the same is true for all the other H-vertices in e_x. Moreover, the change of embedding of e_x does not affect cycle-compatibility of vertices not belonging to e_x, so the new embedding violates fewer cycle-compatibility constraints than the old one, which is a contradiction. This proves that $\mathfrak{P}$ has a compatible embedding. □

Let us remark that there are only finitely many obstructions that may arise from a P-skeleton that lacks a compatible embedding. In fact, if $(G, H, \mathcal{H})$ is a non-planar PEG and if G is a biconnected graph with no K_4-minor (implying that the SPQR-tree of G has no R-nodes), then we may conclude that $(G, H, \mathcal{H})$ contains obstruction 1 or 2, since all the other obstructions contain K_4 as (ordinary) minor.

3.2. R-nodes

Let us now turn to the analysis of R-nodes. As in the case of P-nodes, our goal is to show that if a skeleton $\mathfrak{R}$ of an R-node in the SPQR-tree of G has no compatible embedding, then the corresponding PEG $(G, H, \mathcal{H})$ contains an obstruction. The skeletons of R-nodes have more complicated structure than the skeletons of P-nodes, and accordingly, our analysis is more complicated as well. Similar to the case of P-nodes, we will first show that an R-node of an obstruction-free PEG must have an edge-compatible embedding, and as a second step show that in fact it must also have an edge-compatible embedding that is cycle-compatible.

The skeleton of an R-node is a 3-connected graph. We therefore start with some preliminary observations about 3-connected graphs, which will be used throughout this section. Let $\mathfrak{R}$ be a 3-connected graph with a planar embedding $\mathfrak{R}^+$, let x be a vertex of $\mathfrak{R}$. A vertex y of $\mathfrak{R}$ is *visible from* x if $x \neq y$ and there is a face of $\mathfrak{R}^+$ containing x and y on its boundary. An edge e is visible from x if e is not incident with x and there is a face containing both x and e on its boundary. The vertices and edges visible from x form a cycle in $\mathfrak{R}$. To see this, note that these vertices and edges form a face boundary in $\mathfrak{R}^+ - x$, and every face boundary in an embedding of a 2-connected graph is a cycle. We call this cycle *the horizon of* x.

In the following, we will consider a fixed skeleton $\mathfrak{R}$ of an R-node. Since $\mathfrak{R}$ is 3-connected, it has only two planar embeddings, denoted by $\mathfrak{R}^+$ and $\mathfrak{R}^-$ [15]. Suppose that neither of the two embeddings is compatible. The constraints on the embeddings either stem from a vertex whose incident H-edges project to distinct edges of $\mathfrak{R}$ or from a cycle of $\mathfrak{R}$ that is a projection of an H-cycle whose cycle-compatibility constraints demand exactly one of the two embeddings. Since neither $\mathfrak{R}^+$ nor $\mathfrak{R}^-$ are compatible, there must be at least two such structures, one requiring embedding $\mathfrak{R}^+$, and the other one requiring $\mathfrak{R}^-$. If these structures are far apart in $\mathfrak{R}$, for example, if no vertex of the first structure belongs to the horizon of a vertex of the second structure, it is usually not too difficult to find one of the obstructions. However, if they are close together, a lot of special cases can occur. A significant part of the proof therefore consists in controlling the distance of objects and showing that either an obstruction is present or close objects cannot require different embeddings.

As before, we distinguish two main cases: first, we deal with the situation in which both embeddings of $\mathfrak{R}$ violate edge-compatibility. Next, we consider the situation in which $\mathfrak{R}$ has at least one edge-compatible embedding, but no edge-compatible embedding is cycle-compatible.

3.2.1. $\mathfrak{R}$ has no edge-compatible embedding

Let u be vertex of $\mathfrak{R}$ that violates the edge-compatibility of $\mathfrak{R}^+$, and let v be a vertex violating edge-compatibility of $\mathfrak{R}^-$. If $u = v$, i.e., if a single vertex violates edge-compatibility in both embeddings, the following lemma shows that we can find an occurrence of obstruction 2 in $(G, H, \mathcal{H})$.

Lemma 9. *Assume that an R-node skeleton $\mathfrak{R}$ has a vertex u that violates edge-compatibility in both embeddings of $\mathfrak{R}$. Then $(G, H, \mathcal{H})$ contains obstruction 2.*

Proof. Let $e'_1, \ldots, e'_m$ be the $\mathfrak{R}$-edges incident to u that contain at least one H-edge incident to u. Assume that these edges are listed in their clockwise order around u in the embedding $\mathfrak{R}^+$. Let e_i be an H-edge incident to u projecting into e'_i.

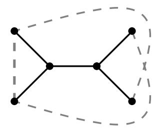


Fig. 7. An example of a minimal wrung PEG that is not a minimal planarity obstruction (it contains obstruction 2).

By Lemma 4, if a triple of edges e'_i, e'_j, e'_k violates edge-compatibility in $\mathfrak{R}^+$, then this violation is demonstrated by the edges e_i, e_j, e_k, i.e., the cyclic order of e_i, e_j and e_k in $\mathcal{H}$ is different from the cyclic order of e'_i, e'_j and e'_k in $\mathfrak{R}^+$.

Choose a largest set $I \subseteq \{1,\dots,m\}$ such that the edges $\{e_i, i \in I\}$ do not contain any violation of edge-compatibility when embedded according to $\mathfrak{R}^+$. Clearly, $3 \leqslant |I|$ because if each triple violated edge-compatibility in $\mathfrak{R}^+$, then $\mathfrak{R}^-$ would be edge-compatible with u. Also $|I| < m$, otherwise $\mathfrak{R}^+$ would be edge-compatible with u.

Choose an index $i \in \{1,\dots,m\}$ not belonging to I. By maximality of I, there are $j,k,\ell \in I$ such that, w.l.o.g., (e_i, e_j, e_k, e_ℓ) appear clockwise in $\mathfrak{R}^+$ and (e_j, e_i, e_k, e_ℓ) appear clockwise in $\mathcal{H}$ (recall that (e_j, e_k, e_ℓ) have the same order in $\mathfrak{R}^+$ and $\mathcal{H}$, by the definition of I).

For $a \in \{1,\dots,m\}$ let x_a be the endpoint of the skeleton edge e'_a different from u. The horizon of u in $\mathfrak{R}^+$ contains two disjoint paths P and Q joining x_i with x_ℓ and x_j with x_k. By obvious contractions we obtain obstruction 2. □

Let us concentrate on the more difficult case when u and v are distinct. To handle this case, we introduce the concept of 'wrung PEGS'. A *wrung* PEG is a PEG $(G, H, \mathcal{H})$ with the following properties.

- G is a subdivision of a 3-connected planar graph, therefore it has two planar embeddings $\mathcal{G}^+$, $\mathcal{G}^-$.
- H has two distinct vertices u and v of degree 3. Any other vertex of H is adjacent to u or v, and any edge of H is incident to u or to v. Hence, H has five or six edges, and at most eight vertices.
- H is not isomorphic to $K_{2,3}$ or to K_4^- (i.e., K_4 with an edge removed). Equivalently, H has at least one vertex of degree 1.
- The embedding $\mathcal{H}$ of H is such that its rotation scheme around u is consistent with $\mathcal{G}^+$ and its rotation scheme around v is consistent with $\mathcal{G}^-$. Note that such an embedding exists due to the previous condition.

Clearly, a wrung PEG is not planar, because neither $\mathcal{G}^+$ nor $\mathcal{G}^-$ is an extension of $\mathcal{H}$. A *minimal wrung* PEG is a wrung PEG that does not contain a smaller wrung PEG as a PEG-minor. A minimal wrung PEG is not necessarily a planarity obstruction—it may contain a smaller non-planar PEG that is not wrung (see Fig. 7). However, it turns out that minimal wrung PEGS are close to being planarity obstructions. The key idea in using wrung PEGS is that they are characterized by being subdivisions of 3-connected graphs, a property that is much easier to control than non-embeddability of PEGS.

The following proposition summarizes the key property of wrung PEGS. In particular, it implies that there are only finitely many minimal wrung PEGS.

Proposition 10. *If $(G, H, \mathcal{H})$ is a minimal wrung* PEG, *then every vertex of G also belongs to H and the graph H is connected.*

Proof. Let $G^\star$ be the 3-connected graph whose subdivision is G. A *subdivision vertex* is a vertex of G of degree 2. A *subdivided edge* is a path in G of length at least two whose every internal vertex is a subdividing vertex and whose endpoints are not subdividing vertices. Therefore, each edge of $G^\star$ either represents an edge of G or a subdivided edge of G.

The proof of the proposition is based on several claims.

Claim 1. *Every subdividing vertex of G is a vertex of H. Every subdivided edge of G contains at most one vertex adjacent to u and at most one vertex adjacent to v. If H is disconnected then G has at most one subdivided edge, which (if it exists) connects u and v and is subdivided by a single vertex.*

If G had a subdividing vertex x not belonging to H, we could contract an edge of G incident to x to get a smaller PEG, which is still wrung.

To see the second part of the claim, note that two vertices adjacent to u in the same subdivided edge would imply the existence of a loop or a multiple edge in $G^\star$.

For the last part of the claim, note that if H is disconnected, then every vertex of H except for u and v has degree 1 in H. If a subdividing vertex adjacent to u were also adjacent to an H-neighbor of v, then the edge between them could be contracted. This proves the claim.

A fundamental tool in the analysis of minimal wrung PEGS is the concept of contractible edges. An edge e in a 3-connected graph F is *contractible* if $F.e$ is also 3-connected, where $F.e$ is the graph obtained from F by contracting e. Note that an edge $e = xy$ in a 3-connected graph F is contractible if and only if $F - \{x, y\}$ is biconnected.

The notion of contractible edges has been intensely studied [10,11], and we are able to use powerful structural theorems that guarantee that any 'sufficiently large' wrung PEGS must contain an edge that can be contracted to yield a smaller wrung Peg. The next fact is a special case of a theorem by Kriesell [10], see also [11, Theorem 3]. We present this result here without proof.

Fact 1. *If F is a 3-connected graph and w a vertex of F that is not incident with any contractible edge and such that $F - w$ is not a cycle, then w is adjacent to four vertices x_1, x_2, y_1, y_2, all having degree 3 in F, which induce two disjoint edges x_1y_1 and x_2y_2 of F, and both these edges are contractible.*

We are now ready to show that every vertex of G also belongs to H. Suppose for a contradiction that G has a vertex w not belonging to H. By Claim 1, w is not a subdivision vertex, so w is also a vertex of $G^\star$. If w were incident to a contractible edge of $G^\star$, we could contract this edge to obtain a smaller wrung Peg. Hence, w is not incident to any contractible edge of $G^\star$. Fix now the four vertices from Fact 1, and let $e_1 = x_1y_1$ and $e_2 = x_2y_2$ be the two contractible edges. Necessarily all the four endpoints of e_1 and e_2 belong to H, otherwise we could contract one of them to get a smaller wrung Peg. Moreover, the edges e_1 and e_2 cannot contain u or v, because their endpoints have degree three and are adjacent to the vertex w not belonging to H. Therefore, each endpoint of e_1 and e_2 is adjacent to either u or v in $G^\star$ (and also in G and in H).

Assume without loss of generality that x_1 is adjacent to u. Then y_1 cannot be adjacent to u, because then u and w would form a separating pair in $G^\star$, hence y_1 is adjacent to v. Analogously, we may assume that x_2 is adjacent to u and y_2 is adjacent to v. The graph H must be connected, otherwise we could contract e_1 or e_2. This means that H, together with e_1 and e_2 and the two edges wx_1 and wx_2 form a subdivision of K_4, and therefore they form a wrung Peg properly contained in $(G, H, \mathcal{H})$. Therefore any vertex of G also belongs to H.

It remains to prove that H is connected. For this we need another concept for dealing with subdivisions of 3-connected graphs. Let F be a 3-connected graph and let $e = xy$ be an edge of F. The *cancellation* of e in F is the operation that proceeds in the three steps: (1) Remove e from F, to obtain $F - e$. (2) If the vertex x has degree 2 in $F - e$, then replace the subdivided edge containing x by a single edge. Do the same for y as well. (3) Simplify the graph obtained from step 2 by removing multiple edges.

Let $F \ominus e$ denote the result of the cancellation of e in F. Note that $F \ominus e$ may contain vertices of degree 2 if they arise in step 3 of the above construction. An edge e is *cancellable* if $F \ominus e$ is 3-connected. It is called *properly cancellable* if it is cancellable, and moreover, the first two steps in the above definition produce a graph without multiple edges.

Claim 2. *A cancellable edge e in a 3-connected graph F is either properly cancellable or contractible.*

Suppose that $e = xy$ is cancellable, but not properly cancellable. We show that it is contractible. Since e is not properly cancellable, one of its endpoints, say x, has degree 3 in F and its two neighbors x' and x'' besides y are connected by an edge. We show that between any pair of vertices a and b of $F - \{x, y\}$ there are two vertex-disjoint paths. In F there exist three vertex-disjoint $a-b$-paths P_1, P_2 and P_3. If two of them avoid x and y then they are also present in $F - \{x, y\}$. Therefore, we may assume that P_1 contains x and P_2 contains y. Then P_1 contains the subpath $x'xx''$ which can be replaced by the single edge $x'x''$. Again at most one of the paths contains vertices of $\{x, y\}$ and therefore we again find two vertex-disjoint a–b-paths in $F - \{x, y\}$. This shows that $F - \{x, y\}$ is biconnected and therefore $e = xy$ is contractible. This concludes the proof of the claim.

Moreover, we need the following result by Holton et al. [6], which we present without proof.

Fact 2. *If F is a 3-connected graph with at least five vertices, then every triangle in F has at least two cancellable edges.*

We proceed with the proof of Proposition 10, and show that H is connected. Suppose for contradiction that H is disconnected, and let H_u and H_v be its two components containing u and v. Let x_1, x_2 and x_3 be the three neighbors of u in H, and y_1, y_2 and y_3 the three neighbors of v. Recall from Claim 1 that G has at most one subdividing vertex, and that the possible subdivided edge connects u and v.

Since $G^\star$ is 3-connected, it has three disjoint edges e_1, e_2 and e_3, each of them connecting a vertex of H_u to a vertex of H_v. At least one of them avoids both u and v. Assume without loss of generality that $e_1 = x_1y_1$ is such an edge. If e_1 were a contractible edge of $G^\star$, we would get a smaller wrung Peg. Therefore the graph $G^\star - \{x_1, y_1\}$ has a cut-vertex w. Note that w is either u or v. Otherwise, $H_u - \{x_1, y_1, w\}$ would be connected, and $H_v - \{x_1, y_1, w\}$ would be connected as well. However, since (by disjointness) at least one of the edges e_2, e_3 avoids w, this implies that $G^\star - \{x_1, y_1, w\}$ would be connected as well, contradicting the choice of w.

So, without loss of generality, $G^\star$ has a separating triplet $\{x_1, y_1, u\}$. Since at least one of the two edges e_2, e_3 avoids this triplet, we see that one of the components of $G^\star - \{x_1, y_1, u\}$ consists of a single vertex $x' \in \{x_2, x_3\}$. Since each vertex in a minimal separator must be adjacent to each of the components separated by the separator, $G^\star$ contains the two edges $x'x_1$ and $x'y_1$. Consequently, x', x_1 and y_1 induce a triangle in $G^\star$ (and in G), and by Fact 2, at least one of the two edges x_1y_1 and $x'y_1$ is cancellable, and by Claim 2, at least one of the two edges is contractible or properly cancellable, contradicting the minimality of $(G, H, \mathcal{H})$.

This completes the proof of Proposition 10. □

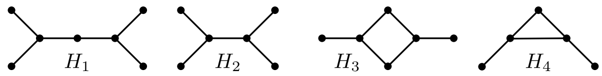


Fig. 8. The four graphs that make the embedded part of any minimal wrung obstruction. Note that H_3 and H_4 have two non-isomorphic embeddings, so there are in total six possibilities how the embedded part of a minimal wrung obstruction may look like.

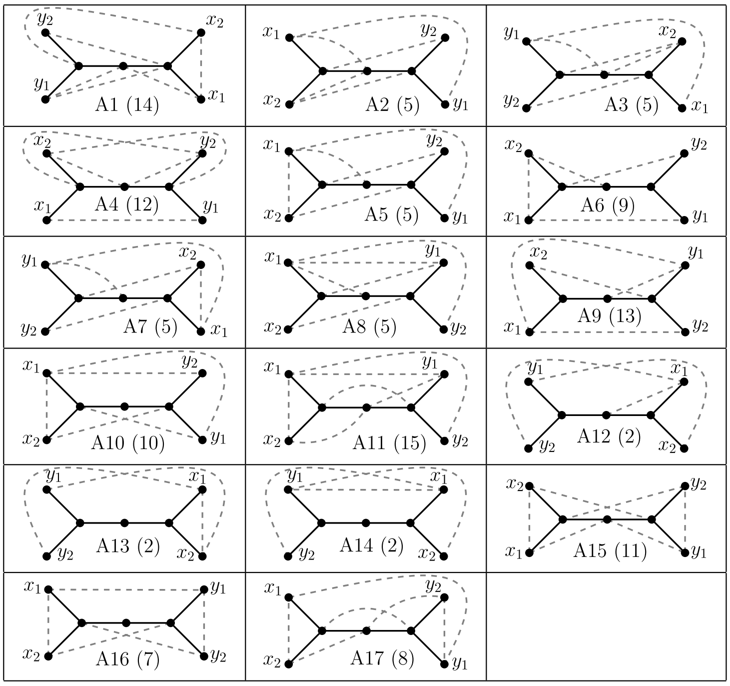


Fig. 9. The minimal wrung Pegs containing H_1 as the embedded part. The numbers in brackets refer to the obstruction contained in the given Peg.

Proposition 10 implies that a minimal wrung Peg $(G, H, \mathcal{H})$ has at most seven vertices, and that the graph H is isomorphic to one of the four graphs H_1, H_2, H_3 and H_4 depicted on Fig. 8. Moreover, in a minimal wrung Peg, the embedded part forms a spanning subgraph.

Our next goal is to show that each minimal wrung Peg contains one of the obstructions of Fig. 1. In view of the previous remarks, to generate all minimal wrung Pegs is a matter of a finite case analysis. Although it is possible to perform such analysis by hand, this is rather tedious and it would make the paper intolerably long. We therefore omit the analysis here and replace it with a simple computer script that simply generates all the minimal wrung Pegs programmatically, using the libraries provided in the Sage package. The code of the script is listed in the Appendix. Note, however, that it is feasible to perform the proof without the help of computer; indeed, the computer-free proof is presented in detail in the arXiv version of this paper [7].

The script we use considers the four graphs H_i of Fig. 8 separately, and for each of them generates all minimal planar spanning supergraphs that are subdivisions of a 3-connected graph. When testing minimality, the script only attempts to remove edges not belonging to H_i, and checks whether this removal generates a smaller wrung Peg.

This procedure is guaranteed to generate all possible minimal wrung Pegs. It may also generate Pegs that are not minimal wrung, e.g., it generates a wrung Peg containing H_1, which can be transformed to a wrung Peg containing H_2 via an H-edge contraction. For our purposes, however, the script is sufficient, since it allows us to verify that each minimal wrung Peg contains an obstruction.

Figs. 9–12 list the wrung Pegs generated by our script. The vertex labels shown on the figures correspond to the labeling used by the script. The embeddings depicted on the figures are consistent with $\mathcal{H}$, and are chosen in order to make the obstructions apparent. Note that the output of the script represents each Peg by a planar embedding of the G-part of the wrung Peg, which is therefore not consistent with the embedding $\mathcal{H}$.

We may now conclude that every wrung Peg contains one of the obstructions from Fig. 1. However, wrung Pegs alone are not sufficient for our analysis of edge-incompatible R-skeletons. We need to introduce another closely related concept of ‘pseudo-wrung’ Peg.

Let $(G, H, \mathcal{H})$ be a wrung Peg, and let u and v be the two vertices of degree 3 in H. Suppose that u and v are connected by an edge of H. Let $(G^\diamond, H^\diamond, \mathcal{H}^\diamond)$ be the Peg obtained from $(G, H, \mathcal{H})$ as follows (see Fig. 13):

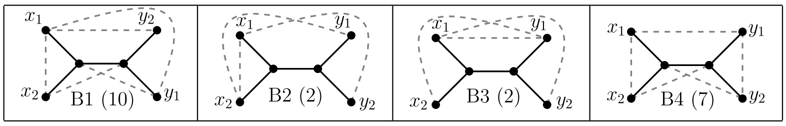


Fig. 10. The minimal wrung PEGs containing H_2 as the embedded part. The numbers in brackets refer to the obstruction contained in the given PEG.

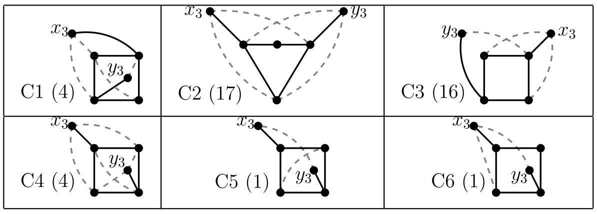


Fig. 11. The minimal wrung PEGs containing H_3 as the embedded part. The numbers in brackets refer to the obstruction contained in the given PEG.

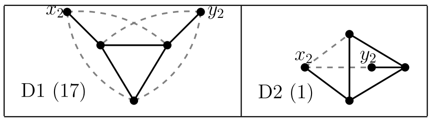


Fig. 12. The minimal wrung PEGs containing H_4 as the embedded part. The numbers in brackets refer to the obstruction contained in the given PEG.

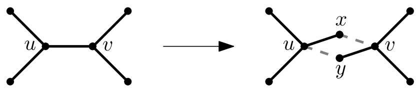


Fig. 13. Modification of a wrung PEG $(G, H, \mathcal{H})$ into a pseudo-wrung PEG $(G^\diamond, H^\diamond, \mathcal{H}^\diamond)$.

- $G^\diamond$ is obtained from G by replacing the edge uv by a fourcycle $uxvy$, where x and y are new vertices not belonging to G,
- $H^\diamond$ is obtained from H by replacing the edge uv by the two edges ux and vy, and
- in the embedding $\mathcal{H}^\diamond$, the new edges ux and vy replace the removed edge uv in the rotation schemes of u and v.

We then say that PEG $(G^\diamond, H^\diamond, \mathcal{H}^\diamond)$ is obtained by *modification* of $(G, H, \mathcal{H})$. We say that a PEG is *pseudo-wrung* if it can be obtained from a wrung PEG by modification.

Lemma 11. *Every pseudo-wrung* PEG *contains one of the obstructions from Fig. 1.*

Proof. Let $(G^\diamond, H^\diamond, \mathcal{H}^\diamond)$ be a pseudo-wrung PEG obtained by modification of a wrung PEG $(G, H, \mathcal{H})$. Let u and v be the two vertices of degree 3 in H (and also in $H^\diamond$). We know that $(G, H, \mathcal{H})$ can be transformed into a minimal wrung PEG by a sequence of PEG-minor operations. Necessarily, the minimal wrung PEG must still contain the edge uv as an embedded edge. Therefore $(G, H, \mathcal{H})$ contains one of the PEGS B1–B4, D1 or D2 from Figs. 10 and 12.

Furthermore, we easily see that if a wrung PEG $(G_2, H_2, \mathcal{H}_2)$ is obtained from $(G, H, \mathcal{H})$ by PEG-minor operations that preserve the edge uv, then an analogous sequence of PEG-minor operations can be applied to the modified pseudo-wrung PEG $(G^\diamond, H^\diamond, \mathcal{H}^\diamond)$, resulting in the pseudo-wrung PEG $(G_2^\diamond, H_2^\diamond, \mathcal{H}_2^\diamond)$ which is the modification of $(G_2, H_2, \mathcal{H}_2)$.

We conclude that $(G^\diamond, H^\diamond, \mathcal{H}^\diamond)$ contains as a PEG-minor one of the six pseudo-wrung PEGS B1$^\diamond$–B4$^\diamond$, D1$^\diamond$ and D2$^\diamond$, obtained by modifications of the PEGS B1–B4, D1 and D2. To prove the lemma, is suffices to verify that each of these six pseudo-wrung PEGS contains an obstruction. This is indeed the case: contracting the edge x_1y_2 in B1$^\diamond$ gives obstruction 6, contracting x_1y_2 in B2$^\diamond$ gives obstruction 5, as does contracting x_2y_1 in B3$^\diamond$. Contracting x_1y_1 in B4$^\diamond$ yields obstruction 6, while D1$^\diamond$ and D2$^\diamond$ are identical to obstructions 6 and 5, respectively. □

Finally, we have all the pieces together to prove the main result of this part.

Proposition 12. *Let $(G, H, \mathcal{H})$ be a biconnected obstruction-free* PEG, *and let $\mathfrak{R}$ be the skeleton of an R-node of the SPQR-tree of G. Then $\mathfrak{R}$ has an edge-compatible embedding.*

Proof. Suppose $\mathfrak{R}$ does not have an edge-compatible embedding. If a single vertex u violates edge-compatibility in both embeddings of $\mathfrak{R}$, we find an obstruction by Lemma 9.

Suppose there are two distinct vertices u and v, each of them violating edge-compatibility of one of the two embeddings of $\mathfrak{R}$. This means that u is incident to three H-edges e_1, e_2, e_3 projecting into distinct $\mathfrak{R}$-edges e'_1, e'_2, e'_3, such that the cyclic order of e_i's in $\mathcal{H}$ coincides with the cyclic order of e'_i's in $\mathfrak{R}^-$, and similarly v is adjacent to H-edges f_1, f_2, f_3 projecting into $\mathfrak{R}$-edges f'_1, f'_2, f'_3, whose order in $\mathfrak{R}^+$ agrees with $\mathcal{H}$.

If all the e'_i and f'_i for $i = 1, 2, 3$ are distinct, then it is fairly easy to see that G must contain a wrung Peg, obtained simply by replacing each edge of $\mathfrak{R}$ with a path of G, chosen in such a way that all the six edges e_i and f_i belong to these paths. Such a choice is always possible and yields a wrung Peg. In particular, this is always the case if u and v are not adjacent in $\mathfrak{R}$.

Suppose now that u and v are connected by an edge g' of $\mathfrak{R}$, and suppose that we have $e'_i = g' = f'_j$ for some i and j. Without loss of generality, suppose that $i = j = 1$. If G has a path from u to v that contains both e_1 and f_1, then we may replace g' by such a path, and replace the remaining edges of $\mathfrak{R}$ by paths as before, to obtain a wrung Peg contained in $(G, H, \mathcal{H})$, and therefore an obstruction.

Let us now assume that G has no path from u to v containing both e_1 and f_1. Then G has a path P_e from u to v that contains e_1, as well as a path P_f from u to v that contains f_1. Necessarily, these paths project into g'. We also know that these two paths are internally disjoint, otherwise we would obtain a single path containing both e_1 and f_1. We now distinguish two cases, depending on whether the edges e_2, e_3, f_2 and f_3 form a fourcycle or not.

Suppose that the four edges e_2, e_3, f_2 and f_3 do not form a fourcycle. We then see that $(G, H, \mathcal{H})$ contains a pseudo-wrung Peg, obtained by contracting $P_e \cup P_f$ into a fourcycle and by replacing all the $\mathfrak{R}$-edges different from g' by paths of G that contain the four edges e_2, e_3, f_2 and f_3. By Lemma 11, $(G, H, \mathcal{H})$ has an obstruction.

It remains to deal with the case when e_2, e_3, f_2 and f_3 form a fourcycle C of H. This means that the $\mathfrak{R}$-edges e'_2, e'_3, f'_2 and f'_3 form a fourcycle in $\mathfrak{R}$. Notice that we now cannot obtain a pseudo-wrung Peg as we did in the previous case: in a wrung Peg, the H-edges cannot form a copy of K_4^-, and therefore in a pseudo-wrung Peg, the H-edges cannot contain a fourcycle. We apply a different argument. Let w and z be the vertices of C distinct from u and v. As $\mathfrak{R}$ is 3-connected, it has a path Q' that connects w and z while avoiding u and v. There is a path Q of G which projects into Q'. The paths Q, P_e and P_f, together with the cycle C, show that $(G, H, \mathcal{H})$ contains obstruction 4. □

3.2.2. $\mathfrak{R}$ has an edge-compatible embedding

Assume now that the embedding $\mathfrak{R}^+$ of the skeleton $\mathfrak{R}$ is edge-compatible but not cycle-compatible. We first give a sketch of our general proof strategy. Our analysis of this situation strongly relies on the concept of C-bridge, which has been previously used by Juvan and Mohar in the study of embedding extensions on surfaces of higher genus [9], and which is also employed (under the name *fragment*) by Demoucron, Malgrange and Pertuiset in their planarity algorithm [2].

Let F be a graph and C a cycle of F. A *C-bridge* is either a chord of C, (i.e., an edge not belonging to C whose vertices are on C) or a connected component of $F - C$, together with all vertices and edges that connect it to C. A vertex of C that is incident to an edge of a C-bridge X is called an *attachment* of X. Let $\mathrm{att}(X)$ denote the set of attachments of X. A bridge that consists of a single edge is *trivial*.

In our argument, we focus on cycles in $\mathfrak{R}$ that are projections of cycles in H. Notice that in this case, any non-trivial bridge in $\mathfrak{R}$ has at least three attachments, because $\mathfrak{R}$ is 3-connected. If $\mathfrak{R}^+$ violates cycle-compatibility, it means that H must contain a cycle C' that projects to a cycle C of $\mathfrak{R}$, and $\mathfrak{R}^+$ has a C-bridge that is embedded on the 'wrong' side of C. We concentrate on the substructures that enforce such 'wrong' position for a given C-bridge, and use them to locate planarity obstructions.

Let us describe the argument in more detail. Suppose again that C' is a cycle of H that projects to a cycle C of $\mathfrak{R}$. Let x be a vertex of H that does not belong to any $\mathfrak{R}$-edge belonging to C. We say that x is *happy* with C', if its embedding in $\mathfrak{R}^+$ does not violate cycle-compatibility with respect to the cycle C', i.e., x is to the left of C' in $\mathcal{H}$ if and only if x is to the left of C in $\mathfrak{R}^+$. Otherwise we say that x is unhappy with C'. We say that a C-bridge B of $\mathfrak{R}$ is happy with C' if there is a vertex x happy with C' that projects into B, and similarly for unhappy bridges. A C-bridge that is neither happy nor unhappy is *indifferent*.

In our analysis of cycle-incompatible skeletons, we establish the following facts.

- With C and C' as above, if a single C-bridge is both happy and unhappy with C', then $(G, H, \mathcal{H})$ contains obstruction 1 or 4 (Lemma 20).
- Let us say that the cycle C' is *happy* if at least one C-bridge is happy with C', and it is *unhappy* if at least one C-bridge is unhappy with C'. If C' is both happy and unhappy, then $(G, H, \mathcal{H})$ contains obstruction 4, obstruction 16, or an alternating-chain obstruction (Lemma 21).
- Assume that the situation described above does not arise. Assume further that C' is an unhappy cycle of $\mathcal{H}$. Then any edge of H incident to a vertex of C' must project into an $\mathfrak{R}$-edge belonging to C, unless $(G, H, \mathcal{H})$ contains obstruction 3 or one of the obstructions from the previous item. Note that this implies, in particular, that the vertices of C impose no edge-compatibility constraints (Lemma 23).
- If C'_1 and C'_2 are two facial cycles of $\mathcal{H}$ whose projection is the same cycle C of $\mathfrak{R}$, then any C-bridge is happy with C'_1 if and only if it is happy with C'_2, unless the graph G is non-planar, or the Peg $(G, H, \mathcal{H})$ contains obstruction 1 (Lemma 24).
- If H contains a happy facial cycle as well as an unhappy one, we obtain obstruction 18 (Lemma 25).

- If H contains an unhappy facial cycle, and if at least one vertex of $\mathfrak{R}$ imposes any non-trivial edge-compatibility constraints, then $(G, H, \mathcal{H})$ contains one of the obstructions 19–22 (Lemma 26).

Note that these facts guarantee that if $(G, H, \mathcal{H})$ is obstruction-free then $\mathfrak{R}$ has a compatible embedding. To see this, assume that $\mathfrak{R}^+$ is an edge-compatible but not cycle-compatible embedding of $\mathfrak{R}$. This means that at least one facial cycle of $\mathcal{H}$ is unhappy. This in turn implies that no cycle may be happy, and no vertex of $\mathfrak{R}$ may impose any edge-compatibility restrictions. Consequently, the embedding $\mathfrak{R}^-$ is compatible. In order to prove the above claims we need some technical machinery, in particular the concept of conflict graph of C-bridges and its properties.

Conflict graph of a cycle and minimality of alternating chains. For a cycle C and two distinct vertices x and y of C, an *arc* of C with endvertices x and y is a path in C connecting x to y. Any two distinct vertices of a cycle determine two arcs. Let u, v, x, y be four distinct vertices of a cycle C. We say that the pair $\{x, y\}$ *alternates with* $\{u, v\}$ if each arc determined by x and y contains exactly one of the two vertices $\{u, v\}$. If U and X are sets of vertices of a cycle C, we say that X *alternates* with U if there are two pairs of vertices $\{u, v\} \subseteq U$ and $\{x, y\} \subseteq X$ that alternate with each other.

Let now F be a graph containing a cycle C. Intuitively, a bridge represents a subgraph, whose internal vertices and edges must all be embedded on the same side of C in any embedding of F. Thus, a C-bridge may be embedded in two possible positions relative to C. Moreover, if two bridges B_1 and B_2 have three common attachments, or if the attachments of B_1 alternate with the attachments of B_2, then in any planar embedding, B_1 and B_2 must appear on different sides of C. This motivates the definition of two types of conflicts between bridges. We say that two C-bridges X and Y of F have a *three-vertex conflict* if they share at least three common attachments, and they have a *four-vertex conflict* if $\operatorname{att}(X)$ alternates with $\operatorname{att}(Y)$. Two C-bridges have a conflict if they have a three-vertex conflict or a four-vertex conflict. This gives rise to a conflict graph of F with respect to C. For a cycle C, define the *conflict graph* K_C to be the graph whose vertices are the C-bridges, and two vertices are connected by an edge of K_C if and only if the corresponding bridges conflict. Define the *reduced conflict graph* K_C^- to be the graph whose vertices are bridges of C, and two bridges are connected by an edge if they have a four-vertex conflict.

As a preparation, we first derive some basic properties of conflict graphs.

Lemma 13. *If F is a planar graph, then for any cycle C of F the conflict graph K_C is bipartite (and hence K_C^- is bipartite as well).*

Proof. In any embedding of F, each C-bridge must be completely embedded on a single side of C. Two conflicting bridges cannot be embedded on the same side of C. □

Consider now the situation when C is a cycle of length at least 4 in a 3-connected graph F. The goal is to show that in this case also the reduced conflict graph K_C^- is connected. To prove this we need some auxiliary lemmas. The first one states that if the attachments of a set of bridges alternate with two given vertices x and y of C, then the set must contain a C-bridge whose attachments alternate with x and y, provided that the set of bridges is connected in the reduced conflict graph K_C^-.

Lemma 14. *Let F be a graph and let C be a cycle in F. Let K be a connected subgraph of the reduced conflict graph K_C^- and let $\operatorname{att}(K)$ be the set of all attachment vertices of the C-bridges in K, that is, $\operatorname{att}(K) = \bigcup_{X \in K} \operatorname{att}(X)$. If $\{x, y\}$ is a pair of vertices of C that alternates with $\operatorname{att}(K)$, then there is a bridge $X \in K$ such that the pair $\{x, y\}$ alternates with $\operatorname{att}(X)$.*

Proof. Let α and β be the two arcs of C with endvertices x and y. Let K_α be the set of C-bridges from K whose all attachments belong to α, and let K_β be the set of bridges from K with all their attachments in β. Note that both K_α and K_β are proper subsets of K, because $\{x, y\}$ alternates with $\operatorname{att}(K)$.

Since no bridge in K_α conflicts with any bridge in K_β, and since K is a connected subgraph in the reduced conflict graph, there must exist a bridge $X \in K$ that belongs to $K \setminus (K_\alpha \cup K_\beta)$. Clearly, X has at least one attachment in the interior of α as well as at least one attachment in the interior of β. Thus, $\operatorname{att}(X)$ alternates with $\{x, y\}$. □

Next, we show that in a 3-connected graph, unless C is a triangle, its reduced conflict graph K_C^- is connected.

Lemma 15. *Let C be a cycle of length at least 4 in a 3-connected graph F. Then the reduced conflict graph K_C^- is connected (and hence K_C is connected as well).*

Proof. We first show that for a cycle C of length at least 4 and a set of C-bridges K that form a connected component in K_C^-, every vertex of C is an attachment of at least one bridge in K.

Claim 3. *Let C be a cycle of length at least four in a 3-connected graph F. Let K be a connected component of the graph K_C^-, and let $\operatorname{att}(K)$ be the set $\bigcup_{X \in K} \operatorname{att}(X)$. Then each vertex of C belongs to $\operatorname{att}(K)$.*

Suppose that some vertices of C do not belong to $\mathrm{att}(K)$. Then there is an arc α of C of length at least 2, whose endvertices belong to $\mathrm{att}(K)$, but none of its internal vertices belongs to $\mathrm{att}(K)$. Let x and y be the endvertices of α. Let β be the other arc determined by x and y. Observe that, since $|\mathrm{att}(K)| \geqslant 3$ in any 3-connected graph, β also has length at least 2.

Since F is 3-connected, $F - \{x, y\}$ is connected, and in particular, there is a C-bridge Y that has at least one attachment u in the interior of the arc α and at least one attachment v in the interior of β. Clearly $Y \notin K$, since Y has an attachment in the interior of α.

Since the pair $\{u, v\}$ alternates with $\{x, y\} \subseteq \mathrm{att}(K)$, Lemma 14 shows that there is a bridge $X \in K$ whose attachments alternate with $\{u, v\}$. Then X and Y have a four-vertex conflict, which is impossible because K is a connected component of K_C^- not containing Y. This finishes the proof of the claim.

We are now ready to prove the lemma. Let K and K' be two distinct connected components of K_C^-. Choose a bridge $X \in K$. Let u and v be any two attachments of X that are not connected by an edge of C. By Claim 3, each vertex of C is in $\mathrm{att}(K')$, so $\mathrm{att}(K')$ alternates with $\{u, v\}$, and hence by Lemma 14, the set K' has a bridge Y whose attachments alternate with the attachments of X. Hence, X and Y have a four-vertex conflict and belong to the same connected component in K_C^-. □

Next, we show that if we have an induced path in the conflict graph, then we can find a corresponding sequence of bridges and pairs of their attachment vertices such that consecutive pairs alternate. This lemma will be the main tool for extracting alternating chains from non-planar PEGS.

Lemma 16. *Let C be a cycle of length at least 4 in a graph F and let P be an induced path with $k \geqslant 2$ vertices in the graph K_C^-. Let $X_1, X_2, \ldots, X_k$ be the vertices of P, with X_i adjacent to X_{i+1} for each $i = 1, \ldots, k-1$. Then for each $i \in \{1, \ldots, k\}$ we may choose a pair of vertices $\{x_i, y_i\} \subseteq \mathrm{att}(X_i)$, such that for each $i = 1, \ldots, k-1$ the pair $\{x_i, y_i\}$ alternates with the pair $\{x_{i+1}, y_{i+1}\}$.*

Proof. For each $j \leqslant k$, select a set $S_j \subseteq \mathrm{att}(X_j)$ in such a way that for each $i < k$ the set S_i alternates with S_{i+1}. Such a selection is possible, e.g., by taking $S_j = \mathrm{att}(X_j)$. Assume now that we have selected $\{S_j \mid j = 1, \ldots, k\}$ so that their total size $\sum_{j \leqslant k} |S_j|$ is as small as possible. We claim that each set S_j consists of a pair of vertices $\{x_j, y_j\}$.

Assume for contradiction that this is not the case. Since obviously each S_j has at least two vertices, assume that for some j we have $|S_j| \geqslant 3$. Clearly, this is only possible for $1 < j < k$. Select a pair of vertices $\{x_{j-1}, y_{j-1}\} \subseteq S_{j-1}$ and a pair of vertices $\{x_{j+1}, y_{j+1}\} \subseteq S_{j+1}$ such that both these pairs alternate with S_j. The sets S_{j-1} and S_{j+1} do not alternate because P was an induced path. Therefore, there is an arc α of C with endvertices $\{x_{j-1}, y_{j-1}\}$ that has no vertex from S_{j+1} in its interior, and similarly there is an arc β with endvertices $\{x_{j+1}, y_{j+1}\}$ and no vertex of S_{j-1} in its interior.

Since both $\{x_{j-1}, y_{j-1}\}$ and $\{x_{j+1}, y_{j+1}\}$ alternate with S_j, there must be a vertex $x_j \in S_j$ that belongs to the interior of α, and a vertex $y_j \in S_j$ belonging to the interior of β. The pair $\{x_j, y_j\}$ alternates with both S_{j-1} and S_{j+1}, contradicting the minimality of our choice of S_j. □

Our next goal is to link the conflict graph with the elements of Ach_k. Recall that an element of Ach_k consists of an H-cycle of length $k+1$ and k edge-disjoint paths $P_1, \ldots, P_k$ such that consecutive pairs have alternating endpoints on C. Moreover, $P_2, \ldots, P_{k-1}$ are single edges, while P_1 and P_k are subdivided by a single isolated H-vertex. Note that for all elements $(G_k, H_k, \mathcal{H}_k)$ of Ach_k, the conflict graph of the unique H_k-cycle forms a path of length k. To establish a link, we consider pairs of a graph and a cycle such that the conflict graph forms a path. Let F be a graph, and let C be a cycle in F. We say that the pair (F, C) forms a *conflict path*, if each C-bridge of F has exactly two attachments and the conflict graph K_C is a path. (Note that if each C-bridge has two attachments, then the conflict graph is equal to the reduced conflict graph.)

Note that if $(G_k, H_k, \mathcal{H}_k)$ is an element of Ach_k and C the unique cycle of H_k, then (G_k, C) forms a conflict path. However, not every conflict path arises this way. Suppose that (F, C) forms a conflict path. Let $e = uv$ be an edge of C. The edge e is called *shrinkable* if no C-bridge attached to u conflicts with any C-bridge attached to v. Note that a shrinkable edge may be contracted without modifying the conflict graph.

Before we can show that the elements of Ach_k are minimal non-planar PEGS, we first need a more technical lemma about conflict paths.

Lemma 17. *Assume that (F, C) forms a conflict path. Then each vertex of C is an attachment for at most two C-bridges.*

Proof. Suppose that (F, C) forms a conflict path, and a vertex $v \in C$ is an attachment of three distinct bridges X, Y and Z. These three bridges do not alternate, so there must be at least five bridges to form a path in K_C. Let x, y and z be the attachments of X, Y and Z different from v. The three vertices x, y and z must be all distinct, because a pair of bridges with the same attachments would share the same neighbors in the conflict graph, which is impossible if the conflict graph is a path with at least five vertices.

Choose an orientation of C and assume that the four attachments appear in the order (v, x, y, z) with respect to this orientation. Let α_{vx}, α_{xy}, α_{yz}, and α_{zx} be the four internally disjoint arcs of C determined by consecutive pairs of these four attachments.

For a subgraph P' of P, let $\operatorname{att}(P')$ denote the set of all the attachments of the bridges that belong to P'. Let P_{xz} be the subpath of K_C that connects X to Z. At least one vertex of $\operatorname{att}(P_{xz})$ must belong to the interior of α_{vx} and at least one vertex of $\operatorname{att}(P_{xz})$ must belong in the interior of α_{zv}. Hence the set $\operatorname{att}(Y)$ alternates with $\operatorname{att}(P_{xz})$ and by Lemma 14, at least one bridge in P_{xz} conflicts with Y. This means that Y is an internal vertex of P_{xz}.

Consider now the graph $P_{xz} - Y$. It consists of two disjoint paths P_x and P_z containing X and Z respectively. We know that P_x has a vertex adjacent to X as well as a vertex adjacent to Y, but no vertex adjacent to Z. Consequently, $\operatorname{att}(P_x)$ contains at least one vertex from the interior of α_{vx} as well as at least one vertex from the interior of α_{yz}. Similarly, $\operatorname{att}(P_z)$ has a vertex from the interior of α_{xy} and from the interior of α_{zv}. Hence, the set $\operatorname{att}(P_x)$ alternates with $\operatorname{att}(P_z)$. Using Lemma 14, we easily deduce that at least one bridge of P_x must conflict with a bridge of P_z, which is a contradiction. □

Next, we show that the attachment vertices on the cycle C of a conflict path (F, C) without shrinkable edges have a structure very similar to that of an alternating chain.

Lemma 18. *Assume that (F, C) forms a conflict path with $k \geqslant 4$ C-bridges. Let $X_1, \dots, X_k$ be the C-bridges, listed in the order in which they appear on the path K_C. Let $\{x_i, y_i\}$ be the two attachments of X_i. Assume that C has no shrinkable edge. Then*

1. *The two attachments $\{x_1, y_1\}$ of X_1 determine an arc α_1 of length 2, and the unique internal vertex z_1 of this arc is an attachment of X_2 and no other bridge.*
2. *The two attachments $\{x_k, y_k\}$ of X_k determine an arc α_k of length 2 different form α_1, and the unique internal vertex z_k of this arc is an attachment of X_{k-1} and no other bridge.*
3. *All the vertices of C other than z_1 and z_k are attachments of exactly two bridges.*

Proof. We know from Lemma 17 that no vertex of C is an attachment of more than two bridges.

Let α and β be the two arcs of C determined by $\{x_1, y_1\}$. The bridges $X_3, \dots, X_k$ do not alternate with X_1, so all their attachments belong to one of the two arcs, say β. The arc α then has only one attachment z_1 in its interior, and this attachment belongs to X_2 and no other bridge. It follows that α has only one internal vertex. This proves the first claim; the second claim follows analogously.

To prove the third claim, note first that any vertex of C must be an attachment of at least one bridge. Suppose that there is a vertex v that is an attachment of only one bridge X_j. Let u and w be the neighbors of v on C. By assumption, both u and w are attachments of at least one bridge that conflicts with X_j.

Assume first, that a single bridge Y conflicting with X_j is attached to both u and w. Since the arc determined by u and w and containing v does not have any other attachment in the interior, this means that Y conflicts only with the bridge X_j. Then $Y \in \{X_1, X_k\}$ and $v \in \{z_1, z_k\}$. Next, assume that the bridge X_{j-1} is attached to u but not to w, and the bridge X_{j+1} is attached to w but not u. We then easily conclude that X_{j-1} conflicts with X_{j+1}, which is a contradiction. □

This directly implies that non-planar Pegs that form a conflict path and do not have shrinkable edges are k-fold alternating chains.

Corollary 1. *Let $(G, H, \mathcal{H})$ be a non-planar* Peg *for which H consists of a single cycle C of length at least 4 and two additional vertices u and v that do not belong to C, such that (G, C) forms a conflict path with bridges $X_1, \dots, X_k$ along the path, each with attachments $\{x_i, y_i\}$. Let further X_i consist of the single edge $x_i y_i$ for $i = 2, \dots, k-1$ and let X_1 consist of $x_1 u y_1$ and X_k of $x_k v y_k$. If C does not contain shrinkable edges then $(G, H, \mathcal{H})$ is an element of* Ach_k.

Proof. The non-planarity of G implies that u and v must be embedded on different sides of C if k is even, and on the same side if k is odd.

Clearly, the graphs G and H have the same vertex set. By assumption, each bridge X_i forms a path P_i, which satisfy the properties for k-fold alternating chains; they have the right lengths and contain the right vertices. Further, since (G, C) forms a conflict path their endpoints alternate in the required way.

Finally, as C has no shrinkable edges, Lemma 18 implies that all vertices of C have degree 4, with the exception of one of the attachments of X_2 and X_{k-1}, which have degree 3. This also implies that the length of the cycle is $k + 1$, and thus $(G, H, \mathcal{H})$ thus is an element of Ach_k. □

We now employ the observations we made so far to show that every element of Ach_k is indeed an obstruction.

Lemma 19. *For each $k \geqslant 3$, every element of* Ach_k *is an obstruction.*

Proof. As observed before, Ach_3 contains a single element, which is the obstruction 4. Assume $k \geqslant 4$, and choose $(G', H', \mathcal{H}') \in \mathrm{Ach}_k$. Let C be the unique cycle of H, and let u and v be the two isolated vertices of H. Observing that $(G', H', \mathcal{H}')$ is not planar is quite straightforward: since no two conflicting bridges can be embedded into the same region

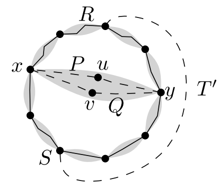


Fig. 14. Illustration of Lemma 20, the bridge embedded in the cycle contains a happy vertex u and an unhappy vertex v that are not connected by a path avoiding x and y. In this case, the PEG contains obstruction 4.

of C, all the odd bridges $X_1, X_3, X_5, \dots$ must be in one region while all the even bridges must be in the other region, and this guarantees that u or v will be on the wrong side of C.

Let us prove that $(G', H', \mathcal{H}')$ is minimal non-planar. The least obvious part is to show that contracting an edge of a cycle C always gives a planar PEG. If the cycle C contained a shrinkable edge $e = xy$, we might contract the edge into a single vertex x_e. After the contraction, the new graph still forms a conflict path, but the vertex x_e is an attachment of at least three bridges, which contradicts Lemma 17. We conclude that C has no shrinkable edge.

By contracting a non-shrinkable edge C, we obtain a new PEG $(G'', H'', \mathcal{H}'')$ where H consists of a cycle C' and two isolated vertices. The conflict graph of C' in G'' is a proper subgraph of the conflict graph of C in G'. In particular, the bridges containing u and v belong to different components of the conflict graph of C'. We may then assign each bridge to one of the two regions of the cycle C', in such a way that the bridges containing u and v are assigned consistently with the embedding $\mathcal{H}''$, and the remaining bridges are assigned in such a way that no two bridges in the same region conflict.

It is easy to see that any collection of C'-bridges that does not have a conflict can be embedded inside a single region of C' without crossing. Thus, $(G'', H'', \mathcal{H}'')$ is planar.

By analogous arguments, we see that removing or relaxing an edge or vertex of H' yields a planar PEG. Contracting an edge incident to u or v yields an planar PEG as well. Thus, $(G', H', \mathcal{H}')$ is an obstruction. □

At least one of the embeddings is edge-compatible. Finally, we use all this preparation to analyze the skeletons of R-nodes. In all the following lemmas we suppose that $(G, H, \mathcal{H})$ is a 2-connected obstruction-free PEG, and that $\mathfrak{R}$ is an R-skeleton of G with at least one edge-compatible embedding $\mathfrak{R}^+$, which we assume to be fixed. We denote this hypothesis (HP1).

Let C be a cycle of $\mathfrak{R}$ that is a projection of a cycle C' of $\mathcal{H}$. Recall that a vertex x of H that does not belong to an edge of C is happy with C' if it is embedded on the correct side of C in $\mathfrak{R}^+$, and that it is unhappy otherwise. Recall further that a C-bridge is happy with C', if it contains a happy vertex, and it is unhappy if it contains an unhappy vertex and that a bridge that is neither happy nor unhappy is indifferent. We first show that a C-bridge cannot be happy and unhappy at the same time.

Lemma 20. *In the hypothesis* (HP1)*, if C is a cycle of $\mathfrak{R}$ that is a projection of a cycle C' of $\mathcal{H}$, then no C-bridge can be both happy and unhappy with C'.*

Proof. Assume a C-bridge X contains a happy vertex u and an unhappy vertex v.

If there exists a G-path from u to v that avoids all the vertices of C', then we obtain obstruction 1. Assume then that there is no such path. This easily implies that the bridge X is a single $\mathfrak{R}$-edge B with two attachments x and y. Fig. 14 shows this situation and illustrates the following steps. Since both u and v are connected to x and to y by a G-path projecting into B, there is a cycle D of G containing both u and v, and which is contained in B. Since every G-path from u to v inside B intersects x or y, we conclude that D can be expressed as a union of two G-paths P and Q from x to y, with $u \in P$ and $v \in Q$.

Similarly, the cycle C of $\mathfrak{R}$ can be expressed as a union of two $\mathfrak{R}$-paths R and S, each with at least one internal vertex. The paths R and S are projections of two H-paths R' and S'. Since $\mathfrak{R}$ is 3-connected, it has a path T that connects an internal vertex of R to an internal vertex of S, and whose internal vertices avoid C. The path T is a projection of a G-path T'. The paths P, Q, R', S' and T' can be contracted to form obstruction 4. □

Recall that a cycle C' in H that projects to a cycle C in $\mathfrak{R}$ is happy, if there is at least one C-bridge that is happy with C' and it is unhappy, if at least one C-bridge is unhappy with C'. Again, as the following lemma shows, cycles cannot be both happy and unhappy at the same time.

Lemma 21. *In the hypothesis* (HP1)*, if C' is a cycle of H whose projection is a cycle C of $\mathfrak{R}$, then C' cannot be both happy and unhappy.*

Proof. Suppose C has a happy bridge X containing a happy vertex u, and an unhappy bridge Y with an unhappy vertex v.

If C is a triangle, then X and Y cannot be chords of C and therefore they have three attachments, each. This implies that they are embedded on different sides of the triangle and all vertices of the triangle are attachments of both X and Y. Since Y is unhappy, it contains a vertex that is prescribed on the same side of C' as X. This yields obstruction 17. Otherwise,

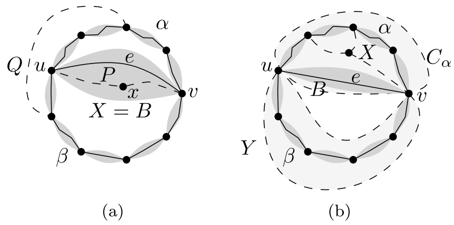


Fig. 15. Illustration of Lemma 22. The edge e is a relevant chord of the shown cycle, and the bridge X is assumed to be unhappy. If the skeleton edge of e also contains an unhappy vertex x, we obtain obstruction 3 (a). Otherwise, the skeleton edge of e together with the arc between its attachments forms a smaller cycle C_α, for which X is still unhappy, but the remainder of the cycle is part of a happy bridge (b), which contradicts the fact that a cycle cannot be both happy and unhappy.

C has length at least 4, and we know that the reduced conflict graph K_C^- is connected by Lemma 15. We find a shortest path $X_1, \dots, X_k$ in K_C^- connecting $X = X_1$ to $Y = X_k$. If the path is a single edge, we obtain obstruction 16. Otherwise we use Lemma 16 to choose for each X_i a pair of attachments $\{x_i, y_i\} \subseteq \mathrm{att}(X_i)$, such that $\{x_i, y_i\}$ alternates with $\{x_{i+1}, y_{i+1}\}$.

Since each C-bridge of the skeleton represents a connected subgraph of G, we know that for every $i = 2, \dots, k-1$ the graph G has a path from x_i to y_i whose internal vertices avoid C' and which projects to the interior of X_i. We also know that there is a G-path Q_1 from x_1 to u, and a G-path R_1 from y_1 to u whose internal vertices avoid C' and which project into X_1. Similarly, there are G-paths Q_k and R_k from x_k to v and from y_k to v, internally disjoint with C' and projecting into X_k. Performing contractions if necessary, we may assume that all these paths are in fact single edges.

Consider the sub-Peg $(G', H', \mathcal{H}')$, where H' consists of the cycle C' and the two vertices u and v, and G has in addition all the edges obtained by contracting the paths defined above. If C' has shrinkable edges, we may contract them, until no shrinkable edges are left. Then we either obtain obstruction 4 (if $k = 3$), or Corollary 1 implies that we have obtained an occurrence of Ach_k for some $k \geqslant 4$. □

Next, we show that it is not possible that one cycle is happy and another one is unhappy. However, this is complicated if the cycles are too close in $\mathfrak{R}$, in particular if they share vertices. Therefore, we first show that an unhappy cycle C' projecting to a cycle C may not have an incident H-edge that does not belong to C. Such an edge e, if it existed, would either be a chord of C', or it would be part of a bridge containing a vertex of H (e.g., the endpoint of e not belonging to C'). The next two lemmas exclude these two cases separately.

In the former case, where e is a chord of C' that hence projects to a chord of C, we also call e a *relevant chord*. Note that if B is an edge of $\mathfrak{R}$ containing a relevant chord, then in an edge-compatible embedding of $\mathfrak{R}$, B must always be embedded on the correct side of C. For practical purposes, such an edge B behaves as a happy bridge, as shown by the next lemma.

Lemma 22. *In the hypothesis* (HP1)*, let C' be a cycle of H that projects to a cycle C of $\mathfrak{R}$. Let e be a relevant chord of C' that projects into an $\mathfrak{R}$-edge B. Then C' cannot be unhappy.*

Proof. Let u and v be the two vertices of e, which are also the two poles of B. Let α' and β' be the two arcs of C' determined by the two vertices u and v, and let α and β be the two arcs of C that are projections of α' and β', respectively. Note that each of the two arcs α and β has at least one internal vertex, otherwise B would not be a chord.

Suppose for contradiction that C has an unhappy bridge X containing an unhappy vertex x. We distinguish two cases, depending on whether B is part of X or not.

First, assume that the bridge X contains the $\mathfrak{R}$-edge B. Then X is a trivial bridge whose only edge is B; see Fig. 15(a). The edge B then contains a G-path P from u to v containing x. The graph G also has a path Q connecting an internal vertex of α to an internal vertex of β and avoiding both u and v. Together, the edge e, the paths P and Q, and the arcs α and β can be contracted to form obstruction 3.

Assume now that the bridge X does not contain B. Consider two H-cycles $C'_\alpha = \alpha' \cup e$ and $C'_\beta = \beta' \cup e$, and their respective projections $C_\alpha = \alpha \cup B$ and $C_\beta = \beta \cup B$. It is not hard to see that the vertex x must be unhappy with at least one of the two cycles C'_α and C'_β. Let us say that X is unhappy with C'_α; see Fig. 15(b). Thus, C_α has at least one unhappy bridge. We claim that C_α also has a happy bridge. Indeed, let Y be the bridge of C_α that contains β. Since β has at least one internal vertex, the bridge Y is not indifferent. The bridge Y must be happy, otherwise the vertices u and v would violate edge-compatibility. This means that C_α has both a happy bridge and an unhappy bridge, contradicting Lemma 21. □

Lemma 23. *In the hypothesis* (HP1)*, let C' be a cycle of H that projects to a cycle C of $\mathfrak{R}$. If C' is unhappy, then every edge of H that is incident to a vertex of C projects into an $\mathfrak{R}$-edge that belongs to C.*

Proof. For contradiction, assume that an edge $e = uv$ of H is incident to a vertex $u \in C$, but projects into an $\mathfrak{R}$-edge $B \notin C$. If v is also a vertex of C, then e is a relevant chord and C may not have any unhappy bridges by Lemma 22. If $v \notin C$, then

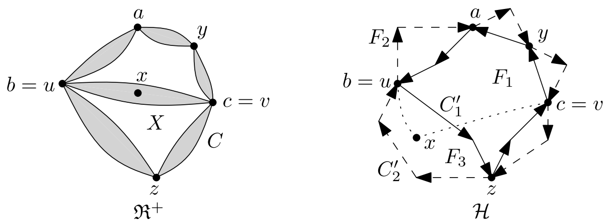


Fig. 16. Illustration of the proof of Lemma 24. In the right part, the solid lines correspond to C_1', the dashed lines represent C_2', and the dotted lines represent the paths from x to the two attachments of X.

v is an internal vertex of a C-bridge, and from edge-compatibility it follows that v is happy with C'. Thus C has both happy and unhappy bridges, contradicting Lemma 21. □

The previous two lemmas show that for an unhappy cycle C' of H projecting to a cycle C of $\mathfrak{R}^+$, no C-bridge contains an H-edge incident to a vertex of C. In particular, the projection of any happy H-cycle is either disjoint from C (that is they are far apart) or it is identical to C. We now exclude the latter case.

Lemma 24. *In the hypothesis* (HP1)*, let C_1' and C_2' be two distinct facial cycles of $\mathcal{H}$, which project to the same* (*undirected*) *cycle C of $\mathfrak{R}$. Then any C-bridge that is happy with C_2' is also happy with C_1'.*

Proof. Let F_1 and F_2 be the faces of $\mathcal{H}$ corresponding to facial cycles C_1' and C_2', respectively.

Suppose for contradiction that at least one C-bridge X is unhappy with C_1' and happy with C_2'. In view of Lemma 20, we may assume that X contains in its interior a vertex $x \in H$, such that x is unhappy with C_1' and happy with C_2'. Refer to Fig. 16.

Suppose that the two facial cycles C_1' and C_2' are oriented in such a way that their corresponding faces are to the left of the cycles. Note that any vertex of C is a common vertex of C_1' and C_2'. This shows that the two facial cycles have at least three common vertices, which implies that they correspond to different faces of $\mathcal{H}$.

Let a, b and c be any three distinct vertices of C, and assume that these three vertices appear in the cyclic order (a, b, c) when the cycle C_1' is traversed according to its orientation. The interior of the face F_2 lies to the right of the cycle C_1', and in particular, the three vertices a, b, c appear in the cyclic order (c, b, a) when the boundary of F_2 is traversed in the orientation of C_2'. Thus, C_1' and C_2' induce opposite orientations of their common projection C. Since x is happy with exactly one of the two cycles C_1' and C_2', it means that in the graph H with embedding $\mathcal{H}$, the two cycles either both have x on their right, or both have x on their left. It is impossible that both facial cycles have x on their left, because the region left of C_1' is disjoint from the region left of C_2'. Hence x is to the right of C_1' and C_2'.

Let H_C be the connected component of H containing the vertices of C, and let $\mathcal{H}_C$ be its embedding inherited from $\mathcal{H}$. By Lemma 23, the bridge X contains no edge of H adjacent to C, so $x \notin H_C$. Let F_3 be the face of $\mathcal{H}_C$ that contains x in its interior. Note that F_3 is distinct from F_1 and F_2, as x is contained in it, which is not the case for F_1 and F_2. All the attachments of the bridge X must belong to the boundary of F_3 (as well as F_1 and F_2), otherwise we would obtain obstruction 1, using the fact that X contains a G-path from x to any attachment of X. If X has at least three attachments, this leads to contradiction, because no three faces of a planar graph can share three common boundary vertices — to see this, imagine inserting a new vertex into the interior of each of the three faces and connecting the new vertices by edges to the three common boundary vertices, to obtain a planar drawing of $K_{3,3}$.

Suppose now that X only has two attachments u and v, which means that X is a trivial bridge. Each of the two arcs of C determined by u and v must have an internal vertex. Let y and z be such internal vertices of the two arcs. To get a contradiction, insert a new vertex w into the interior of face F_1 in $\mathcal{H}$ and connect it by edges to all the four vertices u, v, y, z. Then draw an edge uv inside face F_3 and an edge yz inside F_2. The new edges together with the cycle C_1' form a subdivision of K_5. □

We are now ready to show that $\mathfrak{R}^+$ may not have a happy and an unhappy cycle.

Lemma 25. *In the hypothesis* (HP1)*, let C_1' and C_2' be two cycles of H that project to two distinct cycles C_1 and C_2 of $\mathfrak{R}$. If C_1' is unhappy, then C_2' cannot be happy.*

Proof. Suppose that C_1' is unhappy and C_2' is happy. By Lemma 23, this means that no C_1-bridge may contain an edge of H incident to a vertex of C_1. Consequently, the two cycles C_1 and C_2 are vertex-disjoint. Since $\mathfrak{R}$ is 3-connected, it contains three disjoint paths P_1, P_2 and P_3, each connecting a vertex of C_1 to a vertex of C_2. Each path P_i is a projection of a G-path P_i' connecting a vertex of C_1' to a vertex of C_2'. Note that C_1 is inside a happy bridge of C_2, and C_2 is inside an unhappy bridge of C_1. Thus, contracting the cycles C_1' and C_2' to triangles and contracting the paths P_i' to edges, we obtain obstruction 18. □

The next lemma shows that if any vertex u of $\mathfrak{R}$ that requires the embedding $\mathfrak{R}^+$, then no cycle can be unhappy.

Lemma 26. *In the hypothesis* (HP1)*, assume that H has three edges e_1, e_2 and e_3 that are incident to a common vertex u and project into three distinct $\mathfrak{R}$-edges B_1, B_2 and B_3 of $\mathfrak{R}$. Then no cycle of H that projects to a cycle of $\mathfrak{R}$ can be unhappy.*

Proof. Proceed by contradiction. Assume that there is an unhappy cycle C' of $\mathcal{H}$, which projects to a cycle C of $\mathfrak{R}$. From Lemma 23 it then follows that u does not belong to C, and hence u must belong to an unhappy C-bridge. From the same lemma we also conclude that the vertex u and the three edges e_i belong to a different component of H than the cycle C'.

For $i \in \{1,2,3\}$, suppose that the H-edge e_i connects vertex u to a vertex v_i, and is contained in an $\mathfrak{R}$-edge B_i that connects vertex u to a vertex w_i. These vertices, H-edges and $\mathfrak{R}$-edges are distinct, except for the possibility that $v_i = w_i$.

Let D be the horizon of u in $\mathfrak{R}^+$. The three vertices w_1, w_2 and w_3 split D into three internally disjoint arcs α_{12}, α_{13} and α_{23}, where α_{ij} has endvertices w_i and w_j.

As $\mathfrak{R}$ is 3-connected, it contains three disjoint paths P_1, P_2 and P_3, where P_i connects w_i to a vertex of C. We now distinguish two cases, depending on whether the paths P_i can avoid u or not.

First, assume that it is possible to choose the paths P_i in such a way that all of them avoid the vertex u. We may then add B_i to the path P_i to obtain three paths from u to C, which only share the vertex u. It follows that the graph G contains three paths R'_1, R'_2 and R'_3 from u to C' which are disjoint except for sharing the vertex u, and moreover, each R'_i contains the edge e_i. This yields obstruction 19.

Next, assume that it is not possible to choose P_1, P_2 and P_3 in such a way that all the three paths avoid u.

For $i \in \{1,2,3\}$, let x_i be the last vertex of P_i that belongs to D, assuming the path P_i is traversed from w_i towards C. Let Q_i be the subpath of P_i starting in x_i and ending in a vertex of C (so Q_i is obtained from P_i by removing vertices preceding x_i). Let y_1, y_2 and y_3 be the endvertices of P_1, P_2 and P_3 that belong to C. We may assume that y_i is the only vertex of P_i belonging to C, otherwise we could replace P_i with its proper subpath.

We claim that one of the three arcs α_{12}, α_{13}, and α_{13} must contain all the three vertices x_i, possibly as endvertices. If the vertices x_i did not belong to the same arc, we could connect each x_i to a unique vertex w_j by using the edges of D, and we would obtain three disjoint paths from w_i to C that avoid u. Assume then, without loss of generality, that α_{12} contains all the three vertices x_i.

We may also see that if the cycles C and D share a common vertex y, then y belongs to α_{12}. If not, we could connect w_3 to y by an arc of D that avoids w_1 and w_2, and we could connect w_1 and w_2 to two distinct vertices x_i and x_j by disjoint arcs of D, thus obtaining three disjoint paths from w_i to C avoiding u.

Fix distinct indices $p, q, r \in \{1,2,3\}$ so that the three vertices x_1, x_2 and x_3 are encountered in the order x_p, x_q, x_r when α_{12} is traversed in the direction from w_1 to w_2. Let β be the arc of D contained in α_{12} whose endpoints are x_p and x_r. Clearly x_q is an internal vertex of β.

We claim that at least one internal vertex of β is connected to u by an edge of $\mathfrak{R}$. Assume that this is not the case. Then we may insert into the embedding $\mathfrak{R}^+$ a new edge f connecting x_p and x_r and embedded inside the face of $\mathfrak{R}^+$ shared by x_q and u. Let γ be the arc of C with endvertices y_p and y_r that does not contain y_q. The arc γ, the paths Q_p and Q_r and the edge f together form a cycle in the (multi)graph $\mathfrak{R} \cup \{f\}$. The vertex x_q and the vertex w_q are separated from each other by this cycle. Thus, the path P_q must share at least one vertex with this cycle, but that is impossible, since P_q is disjoint from Q_p, Q_r and γ. We conclude that $\mathfrak{R}$ has an $\mathfrak{R}$-edge B_4 connecting u to a vertex x_4 in the interior of β.

We define three paths R_1, R_2 and R_3 of the graph G as follows. The path R_1 starts in the vertex u, contains the edge $e_1 = uv_1$, proceeds from v_1 to w_1 inside B_1, then goes from w_1 to x_p inside the arc α_{12}, then follows Q_p until it reaches the vertex y_p. Similarly, the path R_2 starts in u, contains the edge e_2, follows from v_2 to w_2 inside B_2, from w_2 to x_r inside α_{12}, and then along Q_r to y_r. The path R_3 starts at the vertex w_3, proceeds towards v_3 inside B_3, then using the edge e_3 it reaches u, proceeds from u to x_4 inside B_4, then from x_4 to x_q inside β, then from x_q towards y_q along Q_q. If any of the three paths R_i contains more than one vertex of C', we truncate the path so that it stops when it reaches the first vertex of C'.

We also define two more paths S_1 and S_2 of G, where each S_i connects the vertex w_i to the vertex w_3 and projects into the arc α_{i3}; see Fig. 17(a) for an illustration of the constructed paths.

Note that the three paths R_i only intersect at the vertex u, a path S_i may only intersect R_j at one of the vertices w_1, w_2 or w_3, and the cycle C' may intersect S_i only in the vertex w_i.

Consider the PEG $(G', H', \mathcal{H}')$ formed by the union of the cycle C', the three paths R_i, and the two paths S_j, where only the cycle C' and the three edges e_1, e_2 and e_3 with their vertices have prescribed embedding, and their embedding is inherited from $\mathcal{H}$.

It can be easily checked that the graph G' is a subdivision of a 3-connected graph, so it has a unique edge-compatible embedding $\mathcal{G}'$. Consider the subgraph $\mathfrak{R}'$ of $\mathfrak{R}$ formed by all the vertices of $\mathfrak{R}$ belonging to G' and all the $\mathfrak{R}$-edges that contain at least one edge of G'. The graph G' is a subdivision of $\mathfrak{R}'$. Thus, the embedding of $\mathfrak{R}'$ inherited from $\mathfrak{R}^+$ must have the same rotation schemes as the embedding $\mathcal{G}'$. Let z_i be the endpoint of R_i belonging to C'. Orient C' so that z_1, z_2, z_3 appear in this cyclic order on C'. Suppose that e_1, e_2 and e_3 appear in this clockwise order in $\mathcal{H}$. Then the four vertices u, v_1, v_2 and v_3 are to the left of C' in $\mathcal{G}'$, and hence also in $\mathfrak{R}^+$. Since the four vertices are in an unhappy C-bridge of $\mathfrak{R}$, they are to the right of C' in $\mathcal{H}'$. This determines $(G', H', \mathcal{H}')$ uniquely.

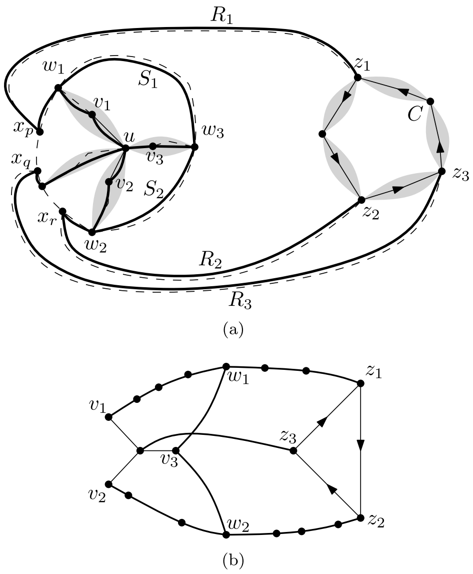


Fig. 17. Illustration of Lemma 26, the paths constructed in the proof (a) and an intermediate step in obtaining one of the obstructions 20, 21 or 22 (b).

We now show that $(G', H', \mathcal{H}')$ contains one of the obstructions 20, 21 or 22. First, we contract each of S_1 and S_2 to a single edge. We also contract the cycle C' to a triangle with vertices z_1, z_2 and z_3. We contract the subpath of R_3 from w_3 to v_3 to a single vertex, and we contract the subpath of R_3 from u to z_3 to a single edge. After reversing the order of the vertices on the cycle to make it happy, we essentially obtain the PEG shown in Fig. 17(b), except that for $i = 1, 2$ it may be that $w_i = v_i$ or $w_i = z_i$, but not both since $v_i \neq z_i$. This is already very close to obstructions 20–22.

To contract R_1, we distinguish two cases. First, assume that w_1 belongs to C'. This means that $z_1 = w_1 \neq v_1$, because we know that v_1 is not in the same component of H as C'. In this case, we contract the subpath of R_1 from v_1 to w_1 to a single edge. On the other hand, if w_1 does not belong to C, we contract the subpath of R_1 from v_1 to w_1 to a single vertex, and we contract the subpath from w_1 to z_1 to a single edge.

The contraction of R_2 is analogous to the contraction of R_1, and it again depends on whether w_2 belongs to C or not. After these contractions are performed, we end up with one of the three obstructions 20, 21, or 22. □

With the lemmas proven so far, we are ready to prove the following proposition.

Proposition 27. *Let $(G, H, \mathcal{H})$ be an obstruction-free* PEG*, with G biconnected. Let $\mathfrak{R}$ be a skeleton of an R-node of the SPQR tree of G. If $\mathfrak{R}$ has at least one edge-compatible embedding, then it has a compatible embedding.*

Proof. Let $\mathfrak{R}^+$ be an edge-compatible embedding. If this embedding is not cycle-compatible, then $\mathcal{H}$ has an unhappy facial cycle C' projecting to a cycle C of $\mathfrak{R}$. The previous lemmas then imply that every facial cycle of $\mathcal{H}$ projecting to a cycle of $\mathfrak{R}$ can only have unhappy or indifferent bridges. Besides, Lemma 26 implies that no vertex u of $\mathfrak{R}$ can be incident to three $\mathfrak{R}$-edges, each of them containing an edge of H incident to u. Hence, the skeleton $\mathfrak{R}$ has no edge-compatibility constraints. Consequently, we may flip the embedding $\mathfrak{R}^+$ to obtain a new embedding that is compatible. □

This concludes our treatment of R-nodes and thus also the proof of the main theorem for biconnected PEGs. We now turn to 1-connected PEGs, i.e., PEGs that are connected but not necessarily biconnected and to disconnected PEGs.

4. Disconnected and 1-connected PEGs

We have shown that a biconnected obstruction-free PEG is planar. We now extend this characterization to arbitrary PEGs. To do this, we will first show that an obstruction-free PEG $(G, H, \mathcal{H})$ is planar if and only if each connected component of G induces a planar sub-PEG. Next, we provide a more technical argument showing that a connected obstruction-free PEG $(G, H, \mathcal{H})$ is planar, if and only if all the elements of a certain collection of 2-connected PEG-minors of $(G, H, \mathcal{H})$ are planar.

4.1. Reduction to G connected

Angelini et al. [1] proved the following lemma.

Lemma 28. *(Cf. Lemma 3.4 in [1].) Let $(G, H, \mathcal{H})$ be a* PEG*. Let $G_1, \ldots, G_t$ be the connected components of G. Let H_i be the subgraph of H induced by the vertices of G_i, and let $\mathcal{H}_i$ be $\mathcal{H}$ restricted to H_i. Then $(G, H, \mathcal{H})$ is planar if and only if*

(1) *each $(G_i, H_i, \mathcal{H}_i)$ is planar, and*
(2) *for each i, for each facial cycle $\vec{C}$ of H_i and for every $j \neq i$, no two vertices of H_j are separated by $\vec{C}$, in other words, all the vertices of H_j are embedded on the same side of C.*

A PEG that does not satisfy the second condition of the lemma must contain obstruction 1. Thus, if Theorem 1 holds for PEGS with G connected, it holds for all PEGS.

4.2. Reduction to G biconnected

Next, we consider connected PEGS, i.e., PEGS $(G, H, \mathcal{H})$ where G is connected. In contrast to planarity of ordinary graphs, it is not in general true that a PEG is planar if and only if each sub-PEG induced by a biconnected component of G is planar. However, for PEGS satisfying some additional assumptions, a similar characterization is possible.

Let $(G, H, \mathcal{H})$ be a connected PEG and let v be a cut-vertex of G. We say that v is H-separating if at least two connected components of $G - v$ contain vertices of H.

Let $(G, H, \mathcal{H})$ be a connected PEG that avoids obstruction 1. Let v be an H-separating cut-vertex of G that does not belong to H. Let x and y be two vertices of H that belong to different connected components of $G - v$, chosen in such a way that there is a path in G connecting x to y whose internal vertices do not belong to H. The existence of such a path implies that x and y share a face F of $\mathcal{H}$, otherwise H would contain a cycle separating x from y, creating obstruction 1. The face F is unique, because x and y belong to distinct components of H. It follows that any planar embedding of G that extends $\mathcal{H}$ must embed the vertex v in the interior of the face F. We define $H' = H \cup v$ and let $\mathcal{H}'$ be the embedding of H' obtained from $\mathcal{H}$ by inserting the isolated vertex v into the interior of the face F. As shown above, any planar embedding of G that extends $\mathcal{H}$ also extends $\mathcal{H}'$. We say that $(G, H', \mathcal{H}')$ is obtained from $(G, H, \mathcal{H})$ by *fixing* the cut-vertex v.

Let $(G, H^+, \mathcal{H}^+)$ be a PEG that is obtained from $(G, H, \mathcal{H})$ by fixing all the H-separating cut-vertices of G not belonging to H. Note that each H^+-separating cut-vertex is also H-separating, and vice versa. A planar embedding of G that extends $\mathcal{H}$ also extends $\mathcal{H}^+$ and in particular, $(G, H, \mathcal{H})$ is planar if and only if $(G, H^+, \mathcal{H}^+)$ is planar. We now show that this operation cannot create a new obstruction in $(G, H^+, \mathcal{H}^+)$.

Lemma 29. *Let $(G, H, \mathcal{H})$ be a connected* PEG *that avoids obstruction 1, and let $(G, H^+, \mathcal{H}^+)$ be the* PEG *obtained by fixing all the H-separating cut-vertices of G. Then $(G, H, \mathcal{H})$ contains a minimal obstruction X if and only if $(G, H^+, \mathcal{H}^+)$ contains X.*

Proof. Since $(G, H, \mathcal{H})$ is a PEG-minor of $(G, H^+, \mathcal{H}^+)$, it suffices to prove that if $(G, H^+, \mathcal{H}^+)$ contains an obstruction $X = (G_X, H_X, \mathcal{H}_X)$ then we can efficiently find the same obstruction in $(G, H, \mathcal{H})$. This clearly holds in the case when H_X does not contain isolated vertices, because then any sequence of deletions, contractions and relaxations that produces X inside $(G, H^+, \mathcal{H}^+)$ will also produce X inside $(G, H, \mathcal{H})$.

Suppose now that H_X contains isolated vertices. Assume first that G_X is 2-connected. Let $G_1, \ldots G_t$ be the 2-connected blocks of G, let H_i be the subgraph of H induced by the vertices of G_i, let $\mathcal{H}_i$ be the embedding of H_i inherited from $\mathcal{H}$, and similarly for H_i^+ and $\mathcal{H}_i^+$. If $(G, H^+, \mathcal{H}^+)$ contains X, then for some i, $(G_i, H_i^+, \mathcal{H}_i^+)$ contains X as well (here we use the fact that each H^+-separating cut-vertex of G belongs to H^+). However, each $(G_i, H_i^+, \mathcal{H}_i^+)$ is a PEG-minor of $(G, H, \mathcal{H})$ — this is because any vertex v of H_i^+ that is not a vertex of H_i is connected to a vertex of H by a path that internally avoids G_i. By contracting all such paths, we obtain a copy of $(G_i, H_i^+, \mathcal{H}_i^+)$ inside $(G, H, \mathcal{H})$. Since $(G_i, H_i^+, \mathcal{H}_i^+)$ contains X, so does $(G, H, \mathcal{H})$.

It remains to deal with the case when X is not 2-connected and H_X contains an isolated vertex. This means that X is obstruction 1. By assumption, $(G, H, \mathcal{H})$ does not contain obstruction 1. Suppose for contradiction that $(G, H^+, \mathcal{H}^+)$ contains obstruction 1. This means that H^+ contains a cycle C and a pair of vertices v and w separated by this cycle, and that there exists a path P of G that connects v and w and has no vertex in common with C.

If v is not a vertex of H, then v is an H-separating cut-vertex. Therefore, there are two vertices x and y of H in distinct components of $G - v$ that both share a face F with v and are connected to v by paths P_x and P_y of G which do not contain any other vertex of H. Since x and y are in distinct components of H, at least one of them, say x, does not belong to the cycle C. Since x shares a face with v, it must be on the same side of C as v. By the same reasoning, the vertex w either belongs to H or there is a vertex $z \in H$ that appears on the same side of C as w and is connected to w by a G-path P_z whose internal vertices do not belong to H. In any case, we find a pair of vertices of H that are separated by C and are connected by a G-path that avoids C. This shows that $(G, H, \mathcal{H})$ contains obstruction 1, which is a contradiction. □

Lemma 29 shows that we can without loss of generality restrict ourselves to PEGS $(G, H, \mathcal{H})$ in which every H-separating cut-vertex belongs to H. For PEGS having this property, we can show that planarity can be reduced to planarity of biconnected components.

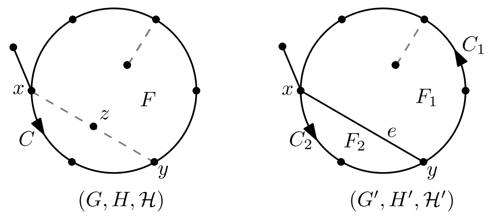


Fig. 18. A reduction rule transforming $(G, H, \mathcal{H})$ into $(G', H', \mathcal{H}')$.

First, we need a definition. Let H be a graph with planar embedding $\mathcal{H}$, let v be a vertex of H, and let H_1 and H_2 be two edge-disjoint subgraphs of H. We say that H_1 and H_2 *alternate* around v in $\mathcal{H}$, if there exist edges $e, e' \in E(H_1)$ and $f, f' \in E(H_2)$ which are all incident with v and appear in the cyclic order (e, f, e', f') in the rotation scheme of v in the embedding $\mathcal{H}$.

The following lemma is analogous to Lemma 3.3 of [1], except that the assumption "every non-trivial H-bridge is local" is replaced with the weaker condition "every H-separating cut-vertex of G is in H". This new assumption is weaker, because a separating cut-vertex not belonging to H necessarily belongs to a non-local H-bridge. However, the proof in [1] uses only this weaker assumption and therefore we have the following lemma.

Lemma 30. *Let $(G, H, \mathcal{H})$ be a connected* Peg *with the property that every H-separating cut-vertex of G is in H. Let $G_1, \dots, G_t$ be the blocks of G, let H_i be the subgraph of H induced by the vertices of G_i and let $\mathcal{H}_i$ be $\mathcal{H}$ restricted to H_i. Then, $(G, H, \mathcal{H})$ is planar if and only if*

(1) *$(G_i, H_i, \mathcal{H}_i)$ is a planar* Peg *for each i,*
(2) *no two distinct graphs H_i and H_j alternate around any vertex of $\mathcal{H}$, and*
(3) *for every facial cycle $\vec{C}$ of $\mathcal{H}$ and for any two vertices x and y of $\mathcal{H}$ separated by $\vec{C}$, any path in G connecting x and y contains a vertex of $\vec{C}$.*

Note that the last two conditions are always satisfied when $(G, H, \mathcal{H})$ avoids obstructions 1 and 2. We can also efficiently test whether the two conditions are satisfied and produce an occurrence of an obstruction when one of the conditions fails. This concludes the proof of Theorem 1.

5. Other minor-like operations

Let us remark that our definition of Peg-minor operations is not the only one possible. In this paper, we preferred to work with a weaker notion of Peg-minors, since this makes the resulting characterization theorem stronger. However, in many circumstances, more general minor-like operations may be appropriate, providing a smaller set of obstructions.

For example, the G-edge contraction rules may be relaxed to allow contractions in more general situations. Here is an example of such a relaxed G-edge contraction rule: given a Peg $(G, H, \mathcal{H})$, assume $e = uv$ is an edge of G but not of H, assume that u and v have a unique common face F of $\mathcal{H}$, and assume furthermore that each of the two vertices is visited only once by the corresponding facial walk of F. If u and v are in distinct components of H, or if the graph H is connected, we embed the edge uv into F and then contract it, resulting in a new Peg $(G', H', \mathcal{H}')$.

It is not hard to see that this relaxed contraction preserves the planarity of a Peg, and that $\mathcal{H}'$ is uniquely determined. It also subsumes the 'complicated G-edge contraction' we introduced. With this stronger contraction rule, most of the exceptional obstructions can be further reduced, leaving only the obstructions 1, 2, 3, 4, 6, 11, 14, 16, and 17, as well as K_5 and $K_{3,3}$. However, even this stronger contraction cannot reduce the obstructions from Ach_k.

To reduce the obstructions to a finite set, we need an operation that can be applied to an alternating chain. We now present an example of such an operation. See Fig. 18.

Suppose that $(G, H, \mathcal{H})$ is a Peg, let F be a face of $\mathcal{H}$, let C be a facial cycle of F oriented in such a way that the interior of F is to the left of C, let x and y be two vertices of C that are not connected by an edge of G, and let z be a vertex of H not belonging to C. Assume that the following conditions hold.

1. The vertex z is adjacent to x and to y in G.
2. The vertex z is embedded to the left of C in the embedding $\mathcal{H}$, and is incident to the face F.
3. Any connected component of H that is embedded to the left of C in $\mathcal{H}$ is connected to a vertex of $C \setminus \{x, y\}$ by an edge of G.
4. Any edge of H that is incident to x or to y and does not belong to C is embedded outside of F (i.e., to the right of C) in $\mathcal{H}$.

We define a new Peg by the following steps.

- Remove vertex z and all its incident edges from G and H.
- Add to G, H and $\mathcal{H}$ a new edge $e = xy$. The edge e is embedded inside F. (Note that the position of e in the rotation schemes of x and y is thus determined uniquely, because of condition 4 above.)
- The edge e splits the face F into two subfaces F_1 and F_2. Let C_1 and C_2 be the facial cycles of F_1 and F_2 such that $C_1 \cup C_2 = C \cup \{e\}$. For any connected component B of H that is embedded to the left of C in H, let w be a vertex of $C \setminus \{x, y\}$ adjacent to a vertex of B. Such a vertex w exists by condition 3 above. If there are more such vertices, we choose one arbitrarily for each B. If w belongs to C_1, then B will be embedded inside F_1, otherwise it will be embedded inside F_2.

Let $(G', H', \mathcal{H}')$ be the resulting PEG. We easily see that if $(G, H, \mathcal{H})$ was planar, then $(G', H', \mathcal{H}')$ is planar as well. In fact, if the vertex z has degree 2 in G, then we may even say that $(G, H, \mathcal{H})$ is planar if and only if $(G', H', \mathcal{H}')$ is planar.

The operation described above allows to reduce each k-fold alternating chain with $k \geqslant 4$ to a smaller non-planar PEG which contains a $(k-1)$-fold alternating chain. It also reduces obstruction 4 to obstruction 3, and obstruction 16 to a PEG that contains obstruction 1. Therefore, when the above operation is added to the permissible minor operations, there will only be a finite number of minimal non-planar PEGs. More precisely, exactly nine minimal non-planar PEGs remain in this case.

Let us point out that the obstructions from the infinite family $\bigcup_{k \geqslant 4} \mathrm{Ach}_k$ only play a role when cycle-compatibility is important. For certain types of PEGs, cycle-compatibility is not a concern. For instance, if the graph H is connected, it can be shown that $(G, H, \mathcal{H})$ is planar if and only if all the skeletons of G have edge-compatible embeddings, and therefore such a PEG is planar if and only if it avoids the finitely many exceptional obstructions.

6. Conclusion

Note that Theorem 1 together with the linear-time algorithm for testing planarity of a PEG [1] immediately implies Theorem 2. In any non-planar instance $I = (G, H, \mathcal{H})$ only linearly many PEG-minor operations are possible. We test each one individually and use the linear-time testing algorithm to check whether the result is non-planar. In this way we either find a smaller non-planar PEG I' resulting from I by one of the operations, or we have found an obstruction, which by Theorem 1 is contained in our list. The running time of this algorithm is at most $O(n^3)$.

In fact, in many cases, as indicated in the paper, obstructions can be found much more efficiently, often in linear time. In particular, the linear-time testing algorithm gives an indication of which property of planar PEGs is violated for a given instance. Is it possible to find an obstruction in a non-planar PEG in linear time? In general, given a fixed PEG $(G, H, \mathcal{H})$, what is the complexity of determining whether a given PEG contains $(G, H, \mathcal{H})$ as PEG-minor? The answer here may depend on the PEG-minor operations we allow.

It is not known whether the results on planar PEGs can be generalized to graphs that have a partial embedding on a higher-genus surface. In fact, even the complexity of recognizing whether a graph partially embedded on a fixed higher-genus surface admits a crossing-free embedding extension is still an open problem.

Appendix A. The code

```
def all_sublists(alist, minsize, maxsize):
    """
    Computes all sublists of alist with length in [minsize , maxsize]
    """
    if (alist==[]):
        if (minsize <= 0 <= maxsize):
            return [[]]
        else:
            return []
    else:
        newlist=alist[:]    #make a copy to avoid modifying alist
        last = newlist.pop()
        rest = all_sublists(newlist, minsize -1, maxsize)
        r1 = [x for x in rest if len(x)>=minsize]
        r2 = [x+[last] for x in rest if len(x)<maxsize]
        return r1+r2

def triconnected_subdiv(g):
    """
    Returns true iff g is subdivision of simple 3-connected graph
    """
    if g.is_clique(): #sage connectivity test can't handle cliques
```

```
        return g.order() > 3
    if g.degree_histogram() [2]==0: # g has no vertices of degree 2?
        return (g.vertex_connectivity() >= 3)
    for (v,deg) in g.degree_iterator(labels=True):
        if (deg<2): # g has a vertex of degree <2?
            return False
        if (deg==2): # g has a subdividing vertex?
            neigh = g[v] # neigh is the list of neighbors of v
            assert len(neigh)== 2
            [x,y]=neigh
            if g.has_edge(x,y):
                return False
            # cancel the subdividing vertex and recurse:
            gnew=copy(g)
            gnew.add_edge(x,y)
            gnew.delete_vertex(v)
            return triconnected_subdiv(gnew)

def good_graph(g):
    """
    Returns true iff g is planar subdivision of a 3-connected graph
    """
    if (g.degree_histogram() [1] >0): #g has a vertex of degree 1?
        return False
    if (not g.is_planar()):
        return False
    if (not triconnected_subdiv(g)):
        return False
    return True

def minimal_good_graph(g):
    """
    Returns true iff g is good
    and removal of any 'G'-edge makes it bad
    """
    if not good_graph(g):
        return False
    for (u,v,lbl) in g.edges(labels=True):
        assert (lbl =='G' or lbl=='H')
        if (lbl == 'G' and g.degree(u)>2 and g.degree(v) >2):
            gnew=copy(g)
            gnew.delete_edge(u,v)
            if triconnected_subdiv(gnew):
                return False   # g is not minimal good
    return True

def supergraphs(h):
    """
    Generates a list of all nonisomorphic minimal
    good graphs having a given graph h as a spanning subgraph
    Edges of h are labelled 'H', other edges are labelled 'G'
    """
    for (u,v) in h.edges (labels=False):
        h.set_edge_label(u,v,'H')   #label edges of h by 'H'
    # list all non-edges of h, add label 'G' to them:
    nonedges_raw=h.complement().edges(labels=False)
    nonedges=[(u,v,'G') for (u,v) in nonedges_raw]
    # max. number of edges to add to h without violating planarity
    maxsize = 3*h.order()-6-h.size()
    # generate a list of lists of non-edges to add into h:
```

```
    candidates = all_sublists(nonedges, 0, maxsize)
    # generate a list of all minimal planar graphs
    # that are a subdivision of a 3-connected planar graph
    # and contain h as a spanning subgraph
    good_graphs = []
    for c in candidates:
        g=copy(h)
        g.add_edges(c)
        if minimal_good_graph(g):
            good_graphs.append(g)
    # remove isomorphic duplicates from good_graphs:
    nonisomorphs = []
    for g in good_graphs:
        for g2 in nonisomorphs:
            if g2.is_isomorphic(g, edge_labels=True):
                break
        else: #'else' executes when for-cycle exits without break
            nonisomorphs.append(g)
    return nonisomorphs

def display_list(glist):
    """
    Displays each graph in the list glist,
    Uses color coding to distinguish edges labelled 'H' and 'G'
    """
    for g in glist:
        hedges = []
        gedges = []
        for (x,y,lbl) in g.edges(labels=True):
            if (lbl=='H'):
                hedges.append( (x,y) )
            else:
                assert (lbl == 'G')
                gedges.append((x,y))
        g.layout(save_pos=True, layout="planar")
        g.show(edge_colors = { 'red':hedges, 'blue':gedges })

# The four possible H-components of a minimal wrung obstruction.
# For each of them, we compute the minimal
# planar 3-connected subdivisions
# containing H as a spanning subgraph

h1=Graph({ 'u':[ 'x1' , 'x2' , 'w'], 'v':['y1' , 'y2' , 'w']})
h1list= supergraphs(h1)

h2=Graph({ 'u':[ 'x1' , 'x2' , 'v'] , 'v':[ 'y1' , 'y2' , 'u']})
h2list = supergraphs(h2)

h3 = Graph({ 'u':[ 'x1' , 'x2' , 'x3' ], 'v':[ 'x1' , 'x2' , 'y3'] })
h3list = supergraphs(h3)

h4 = Graph({ 'u':[ 'v' , 'x1' , 'x2'], 'v':[ 'u' , 'x1' , 'y2'] })
h4list = supergraphs(h4)

display_list(h1list + h2list + h3list + h4list)
```

References

[1] P. Angelini, G. Di Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, I. Rutter, Testing planarity of partially embedded graphs, in: Proceedings 21st ACM–SIAM Symposium on Discrete Algorithms (SODA'10), SIAM, 2010, pp. 202–221.
[2] G. Demoucron, Y. Malgrange, R. Pertuiset, Reconnaissance et construction de représentations planaires topologiques, Rev. Franc. Rech. Oper. 8 (1964) 33–34.
[3] G. Di Battista, R. Tamassia, On-line maintenance of triconnected components with SPQR-trees, Algorithmica 15 (4) (1996) 302–318.
[4] G. Di Battista, R. Tamassia, On-line planarity testing, SIAM J. Comput. 25 (5) (1996) 956–997.

[5] C. Gutwenger, P. Mutzel, A linear time implementation of SPQR-trees, in: Graph Drawing (GD'00), in: LNCS, vol. 1984, 2001, pp. 77–90.
[6] D.A. Holton, B. Jackson, A. Saito, N.C. Wormald, Removable edges in 3-connected graphs, J. Graph Theory 14 (4) (1990) 465–473.
[7] V. Jelínek, J. Kratochvíl, I. Rutter, A Kuratowski-type theorem for planarity of partially embedded graphs, CoRR, arXiv:1204.2915, 2012.
[8] M. Jünger, M. Schulz, Intersection graphs in simultaneous embedding with fixed edges, J. Graph Algorithms Appl. 13 (2) (2009) 205–218.
[9] M. Juvan, B. Mohar, 2-restricted extensions of partial embeddings of graphs, European J. Comb. 26 (3–4) (2005) 339–375.
[10] M. Kriesell, Contractible subgraphs in 3-connected graphs, J. Comb. Theory Ser. B 80 (2000) 32–48.
[11] M. Kriesell, A survey on contractible edges in graphs of a prescribed vertex connectivity, Graphs and Combinatorics 18 (2002) 1–30.
[12] K. Kuratowski, Sur le problème des courbes gauches en topologie, Fund. Math. 15 (1930) 217–283.
[13] P. Mutzel, The SPQR-tree data structure in graph drawing, in: Proceedings of the 30th International Conference on Automata, Languages and Programming (ICALP'03), Springer-Verlag, Berlin, Heidelberg, 2003, pp. 34–46.
[14] K. Wagner, Über eine Eigenschaft der ebenen Komplexe, Mathematische Annalen 114 (1937) 570–590, http://dx.doi.org/10.1007/BF01594196.
[15] H. Whitney, Congruent graphs and the connectivity of graphs, Amer. J. Math. 43 (1932) 150–168.

Appendix G

Planar Embeddings with Small and Uniform Faces

Planar Embeddings with Small and Uniform Faces

Giordano Da Lozzo[1( )], Vít Jelínek[2], Jan Kratochvíl[3], and Ignaz Rutter[3,4]

[1] Department of Engineering, Roma Tre University, Rome, Italy
dalozzo@dia.uniroma3.it
[2] Computer Science Institute, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
jelinek@iuuk.mff.cuni.cz
[3] Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
honza@kam.mff.cuni.cz
[4] Faculty of Informatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
rutter@kit.edu

Abstract. Motivated by finding planar embeddings that lead to drawings with favorable aesthetics, we study the problems MinMaxFace and UniformFaces of embedding a given biconnected multi-graph such that the largest face is as small as possible and such that all faces have the same size, respectively. We prove a complexity dichotomy for MinMaxFace and show that deciding whether the maximum is at most k is polynomial-time solvable for $k \leq 4$ and NP-complete for $k \geq 5$. Further, we give a 6-approximation for minimizing the maximum face in a planar embedding. For UniformFaces, we show that the problem is NP-complete for odd $k \geq 7$ and even $k \geq 10$. Moreover, we characterize the biconnected planar multi-graphs admitting 3- and 4-uniform embeddings (in a k-uniform embedding all faces have size k) and give an efficient algorithm for testing the existence of a 6-uniform embedding.

1 Introduction

While there are infinitely many ways to embed a connected planar graph into the plane without edge crossings, these embeddings can be grouped into a finite number of equivalence classes, so-called *combinatorial embeddings*, where two embeddings are *equivalent* if the clockwise order around each vertex is the same. Many algorithms for drawing planar graphs require that the input graph is provided

Work by Giordano Da Lozzo was supported in part by the Italian Ministry of Education, University, and Research (MIUR) under PRIN 2012C4E3KT national research project "AMANDA – Algorithmics for MAssive and Networked DAta". Work by Jan Kratochvíl and Vít Jelínek was supported by the grant no. 14-14179S of the Czech Science Foundation GAČR. Ignaz Rutter was supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD).

H.-K. Ahn and C.-S. Shin (Eds.): ISAAC 2014, LNCS 8889, pp. 633–645, 2014.
DOI: 10.1007/978-3-319-13075-0_50

together with a combinatorial embedding, which the algorithm preserves. Since the aesthetic properties of the drawing often depend critically on the chosen embedding, e.g. the number of bends in orthogonal drawings, it is natural to ask for a planar embedding that will lead to the best results.

In many cases the problem of optimizing some cost function over all combinatorial embeddings is NP-complete. For example, it is known that it is NP-complete to test the existence of an embedding that admits an orthogonal drawing without bends or an upward planar embedding [10]. On the other hand, there are efficient algorithms for minimizing various measures such as the radius of the dual [1,2] and attempts to minimize the number of bends in orthogonal drawings subject to some restrictions [3,4,6].

Usually, choosing a planar embedding is considered as deciding the circular ordering of edges around vertices. It can, however, also be equivalently viewed as choosing the set of facial cycles, i.e., which cycles become face boundaries. In this sense it is natural to seek an embedding whose facial cycles have favorable properties. For example, Gutwenger and Mutzel [12] give algorithms for computing an embedding that maximizes the size of the outer face. The most general form of this problem is as follows. The input consists of a graph and a cost function on the cycles of the graph, and we seek a planar embedding where the sum of the costs of the facial cycles is minimum. This general version of the problem has been introduced and studied by Mutzel and Weiskircher [14]. Woeginger [15] shows that it is NP-complete even when assigning cost 0 to all cycles of size up to k and cost 1 for longer cycles. Mutzel and Weiskircher [14] propose an ILP formulation for this problem based on SPQR-trees.

In this paper, we focus on two specific problems of this type, aimed at reducing the visual complexity and eliminating certain artifacts related to face sizes from drawings. Namely, large faces in the interior of a drawing may be perceived as holes and consequently interpreted as an artifact of the graph. Similarly, if the graph has faces of vastly different sizes, this may leave the impression that the drawn graph is highly irregular. However, rather than being a structural property of the graph, it is quite possible that the artifacts in the drawing rather stem from a poor embedding choice and can be avoided by choosing a more suitable planar embedding.

We thus propose two problems. First, to avoid large faces in the drawing, we seek to minimize the size of the largest face; we call this problem MinMaxFace. Second, we study the problem of recognizing those graphs that admit perfectly uniform face sizes; we call this problem UniformFaces. Both problems can be solved by the ILP approach of Mutzel and Weiskircher [14] but were not known to be NP-hard.

Our Contributions. First, in Section 3, we study the computational complexity of MinMaxFace and its decision version k-MinMaxFace, which asks whether the input graph can be embedded such that the maximum face size is at most k. We prove a complexity dichotomy for this problem and show that k-MinMaxFace is polynomial-time solvable for $k \leq 4$ and NP-complete for $k \geq 5$. Our hardness result for $k \geq 5$ strengthens Woeginger's result [15], which states that it is NP-complete to minimize the number of faces of size greater than k for $k \geq 4$, whereas our reduction shows that it is in fact NP-complete to decide whether such faces can be completely avoided. Furthermore, we give an efficient 6-approximation for MinMaxFace.

Second, in Section 4, we study the problem of recognizing graphs that admit perfectly uniform face sizes (UniformFaces), which is a special case of k-MinMaxFace. An embedding is *k-uniform* if all faces have size k. We characterize the biconnected multi-graphs admitting a k-uniform embedding for $k = 3, 4$ and give an efficient recognition algorithm for $k = 6$. Finally, we show that for odd $k \geq 7$ and even $k \geq 10$, it is NP-complete to decide whether a planar graph admits a k-uniform embedding.

For space limitations, proofs are sketched or omitted; refer to [5] for full proofs.

2 Preliminaries

A graph $G = (V, E)$ is *connected* if there is a path between any two vertices. A *cutvertex* is a vertex whose removal disconnects the graph. A separating pair $\{u, v\}$ is a pair of vertices whose removal disconnects the graph. A connected graph is *biconnected* if it does not have a cutvertex and a biconnected graph is *3-connected* if it does not have a separating pair. Unless specified otherwise, throughout the rest of the paper we will consider graphs without loops, but with possible multiple edges.

We consider *st*-graphs with two special *pole* vertices s and t. The family of *st*-graphs can be constructed in a fashion very similar to series-parallel graphs. Namely, an edge st is an *st*-graph with poles s and t. Now let G_i be an *st*-graph with poles s_i, t_i for $i = 1, \dots, k$ and let H be a planar graph with two designated adjacent vertices s and t and $k + 1$ edges $st, e_1, \dots, e_k$. We call H the *skeleton* of the composition and its edges are called *virtual edges*; the edge st is the *parent edge* and s and t are the poles of the skeleton H. To compose the G_i's into an *st*-graph with poles s and t, we remove the edge st from H and replace each e_i by G_i for $i = 1, \dots, k$ by removing e_i and identifying the poles of G_i with the endpoints of e_i. In fact, we only allow three types of compositions: in a *series composition* the skeleton H is a cycle of length $k + 1$, in a parallel composition H consists of two vertices connected by $k + 1$ parallel edges, and in a *rigid composition* H is 3-connected.

For every biconnected planar graph G with an edge st, the graph $G - st$ is an *st*-graph with poles s and t [7]. Much in the same way as series-parallel graphs, the *st*-graph $G - st$ gives rise to a (de-)composition tree $\mathcal{T}$ describing how it can be obtained from single edges. The nodes of $\mathcal{T}$ corresponding to edges, series, parallel, and rigid compositions of the graph are *Q-, S-, P-,* and *R-nodes*, respectively. To obtain a composition tree for G, we add an additional root Q-node representing the edge st. We associate with each node μ the skeleton of the composition and denote it by $\mathrm{skel}(\mu)$. For a Q-node μ, the skeleton consists of the two endpoints of the edge represented by μ and one real and one virtual edge between them representing the rest of the graph. For a node μ of $\mathcal{T}$, the *pertinent graph* pert(μ) is the subgraph represented by the subtree with root μ. For a virtual edge ε of a skeleton skel(μ), the *expansion graph* of ε is the pertinent graph pert(μ') of the neighbor μ' corresponding to ε when considering $\mathcal{T}$ rooted at μ.

The *SPQR-tree* of G with respect to the edge st, originally introduced by Di Battista and Tamassia [7], is the (unique) smallest decomposition tree $\mathcal{T}$ for G. Using a different edge $s't'$ of G and a composition of $G - s't'$ corresponds to rerooting $\mathcal{T}$ at the node representing $s't'$. It thus makes sense to say that $\mathcal{T}$ is the SPQR-tree of G. The SPQR-tree of G has size linear in G and can be computed in linear time [11]. Planar embeddings of G correspond bijectively to planar embeddings of all skeletons of $\mathcal{T}$; the choices are the orderings of the parallel edges in P-nodes and the embeddings of the R-node skeletons, which are unique up to a flip. When considering rooted SPQR-trees, we assume that the embedding of G is such that the root edge is incident to the outer face, which is equivalent to the parent edge being incident to the outer face in each skeleton. We remark that in a planar embedding of G, the poles of any node μ of $\mathcal{T}$ are incident to the outer face of *pert*(μ). Hence, in the following we only consider embeddings of the pertinent graphs with their poles lying on the same face.

3 Minimizing the Maximum Face

In this section we present our results on MinMaxFace. We first strengthen the result of Woeginger [15] and show that k-MinMaxFace is NP-complete for $k \geq 5$ and then present efficient algorithms for $k = 3, 4$. In particular, the hardness result also implies that the problem MinMaxFace is NP-hard. Finally, we give an efficient 6-approximation for MinMaxFace on biconnected graphs. Recall that we allow graphs to have multiple edges.

Theorem 1. k-MinMaxFace *is NP-complete for any* $k \geq 5$.

Sketch of Proof. Clearly, the problem is in NP. We sketch hardness for $k = 5$. Our reduction is from Planar 3-Sat where every variable occurs at most three times, which is NP-complete [8, Lemma 2.1]. Note that we can assume without loss of generality that for each variable both literals appear in the formula and, by replacing some variables by their negations, we can assume that a variable with three occurrences occurs twice as a positive literal. Let φ be such a formula.

We construct gadgets where some of the edges are in fact two parallel paths, one consisting of a single edge and one of length 2 or 3. The ordering of these paths then decides which of the faces incident to the gadget edge is incident to a path of length 1 and which is incident to a path of length 2 or 3; see Fig. 1a. Due to this use, we also call these gadgets (1, 2)- and (1, 3)-edges, respectively.

The gadget for a variable with three occurrences is shown in Fig. 1b. The central (1, 3)-edge (*variable edge*) decides the truth value of the variable. Depending on its flip either the positive literal edges or the negative literal edge must be

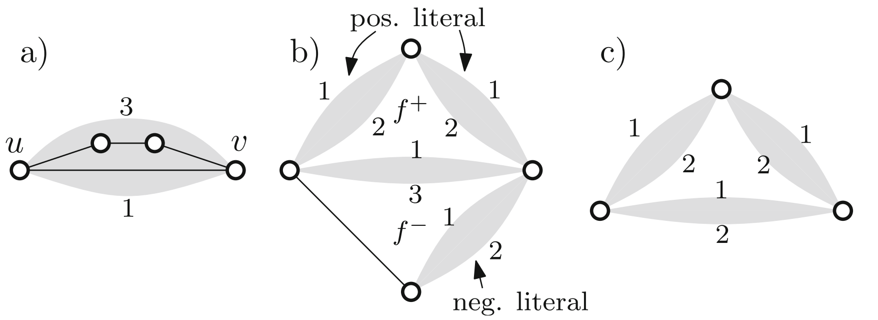


Fig. 1. Illustration of the gadgets for the proof of Theorem 1. (a) A $(1, 3)$-edge. (b) A variable gadget for a variable that occurs twice as a positive literal and once as a negative literal. Changing the flip of the $(1, 3)$-edge in the middle (variable edge) forces flipping the upper two literal edges. (c) A clause gadget for a clause of size 3.

embedded such that they have a path of length 2 in the outer face, which corresponds to a literal with truth value `false`. Figure 1c shows a clause gadget with three incident literal variables. Its inner face has size at most 5 if not all incident $(1, 2)$-edges transmit value `false`. Clauses of size 2 and variables occurring only twice work similarly.

We now construct a graph G_φ by replacing in the plane variable–clause graph of φ each variable and each clause by a corresponding gadget and identifying $(1, 2)$-edges that represent the same variable, taking into account the embedding of the variable–clause graph. Finally, we arbitrarily triangulate all faces that are not inner faces of gadgets. Then the only embedding choices are the flips of the $(1, 2)$- and $(1, 3)$-edges. We claim that φ is satisfiable if and only if G_φ has a planar embedding where every face has size at most 5. $\square$

3.1 Polynomial-Time Algorithm for Small Faces

Next, we show that k-MinMaxFace is polynomial-time solvable for $k = 3, 4$. Note that, if the input graph is simple, the problem for $k = 3$ is solvable if and only if the input graph is maximal planar. A bit more work is necessary if we allow parallel edges.

Let G be a biconnected planar graph. We devise a dynamic program on the SPQR-tree $\mathcal{T}$ of G. Let $\mathcal{T}$ be rooted at an arbitrary Q-node and let μ be a node of $\mathcal{T}$. We call the clockwise and counterclockwise paths connecting the poles of μ along the outer face the *boundary paths* of $\mathrm{pert}(\mu)$. We say that an embedding of $\mathrm{pert}(\mu)$ has type (a, b) if and only if all its inner faces have size at most k and its boundary paths have length a and b, respectively. Such an embedding is also called an (a, b)-embedding. We assume that $a \leq b$.

Clearly, each of the two boundary paths of $\mathrm{pert}(\mu)$ in an embedding $\mathcal{E}_\mu$ of type (a, b) will be a proper subpath of the boundary of a face in any embedding of G where the embedding of $\mathrm{pert}(\mu)$ is $\mathcal{E}_\mu$. Hence, when seeking an embedding where all faces have size at most k, we are only interested in the embedding $\mathcal{E}_\mu$ if $1 \leq a \leq b \leq k - 1$. We define a partial order on the embedding types by

$(a', b') \preceq (a, b)$ if and only if $a' \leq a$ and $b' \leq b$. Replacing an (a, b)-embedding $\mathcal{E}_\mu$ of pert(μ) by (a reflection of) an (a', b')-embedding $\mathcal{E}'_\mu$ with $(a', b') \preceq (a, b)$ does not create faces of size more than k; all inner faces of $\mathcal{E}'_\mu$ have size at most k by assumption, and the only other faces affected are the two faces incident to the two boundary paths of $\mathcal{E}'_\mu$, whose length does not increase. We thus seek to compute for each node μ the minimal pairs (a, b) for which it admits an (a, b)-embedding. We remark that pert(μ) can admit an embedding of type $(1, b)$ for some value of b only if μ is either a P-node or a Q-node.

Theorem 2. *3-MinMaxFace can be solved in linear time for biconnected graphs.*

We now deal with the case $k = 4$, which is similar but more complicated. The relevant types are $(1, 1), (1, 2), (1, 3), (2, 2), (2, 3)$, and $(3, 3)$. We note that precisely the two elements $(2, 2)$ and $(1, 3)$ are incomparable with respect to $\preceq$. Thus, it seems that, rather than computing only the single smallest type for which each pertinent graph admits an embedding, we are now forced to find all minimum pairs for which the pertinent graph admits a corresponding embedding. However, by the above observation, if a pertinent graph pert(μ) admits a $(1, 3)$-embedding, then μ must be a P-node. However, if the parent of μ is an S-node or an R-node, then using a $(1, 3)$-embedding results in a face of size at least 5. Thus, such an embedding can only be used if the parent is the root Q-node. If there is the choice of a $(2, 2)$-embedding in this case, it can of course also be used at the root. Therefore, we can mostly ignore the $(1, 3)$-case and consider the linearly ordered embedding types $(1, 1), (1, 2), (2, 2), (2, 3)$ and $(3, 3)$. The running time stems from the fact that, for an R-node, we need to find a matching between the virtual edges whose expansion graphs admit a $(1, 2)$-embedding and the incident triangular faces of the skeleton.

Theorem 3. *4-MinMaxFace can be solved in $O(n^{1.5})$ time for biconnected graphs.*

3.2 Approximation Algorithm

In this section, we present a constant-factor approximation algorithm for the problem of minimizing the largest face in an embedding of a biconnected graph G. We omit the correctness proofs and some of the technical details.

We again solve the problem by dynamic programming on the SPQR-tree of G.

Let G be a biconnected planar graph, and let $\mathcal{T}$ be its SPQR-tree, rooted at an arbitrary Q-node. Let μ be a node of $\mathcal{T}$. We also include the parent edge in the embedding of skel(μ), by drawing it in the outer face. In such an embedding, the two faces incident to the parent edge are called *the outer faces*; the remaining faces are *inner faces.*

Recall that an (a, b)-*embedding* of pert(μ) is an embedding whose boundary paths have lengths a and b, where we always assume that $a \leq b$. We say that an (a, b)-embedding of pert(μ) is *out-minimal* if for any (a', b')-embedding of pert(μ), we have $a \leq a'$ and $b \leq b'$. Note that an out-minimal embedding need not exist; e.g., pert(μ) may admit a $(2,4)$-embedding and a $(3,3)$-embedding, but no (a,b)-embedding with $a \leq 2$ and $b \leq 3$. We will later show, however, that such a situation can only occur when μ is an S-node.

Let $\mathrm{OPT}(G)$ denote the smallest integer k such that G has an embedding whose every face has size at most k. For a node μ of $\mathcal{T}$, we say that an embedding of pert(μ) is *c-approximate*, if each inner face of the embedding has size at most $c \cdot \mathrm{OPT}(G)$.

Call an embedding of pert(μ) *neat* if it is out-minimal and 6-approximate. The main result of this section is the next proposition.

Proposition 1. *Let G be a biconnected planar graph with SPQR tree $\mathcal{T}$, rooted at an arbitrary Q-node. Then the pertinent graph of every Q-node, P-node or R-node of $\mathcal{T}$ has a neat embedding, and this embedding may be computed in polynomial time.*

Since the pertinent graph of the root of $\mathcal{T}$ is the whole graph G, the proposition implies a polynomial 6-approximation algorithm for minimizing the largest face.

Our proof of Proposition 1 is constructive. Fix a node μ of $\mathcal{T}$ which is not an S-node. We now describe an algorithm that computes a neat embedding of pert(μ), assuming that neat embeddings are available for the pertinent graphs of all the descendant nodes of μ that are not S-nodes. We distinguish cases based on the type of the node μ. We here only present the two difficult cases, when μ is a P-node or an R-node.

P-nodes. Suppose that μ is a P-node with k child nodes $\mu_1, \ldots, \mu_k$, represented by k skeleton edges $e_1, \ldots, e_k$. Let G_i be the expansion graph of e_i. We construct the *expanded skeleton* skel$^*(\mu)$ as follows: if for some i the child node μ_i is an S-node whose skeleton is a path of length m, replace the edge e_i by a path of length m, whose edges correspond in a natural way to the edges of skel(μ_i).

Every edge e' of the expanded skeleton corresponds to a node μ' of $\mathcal{T}$ which is a child or a grand-child of μ. Moreover, μ' is not an S-node, and we may thus assume that we have already computed a neat embedding for pert(μ'). Note that pert(μ') is the expansion graph of e'.

For each $i \in \{1, \ldots, k\}$ define ℓ_i to be the smallest value such that G_i has an embedding with a boundary path of length ℓ_i. We compute ℓ_i as follows: if μ_i is not an S-node, then we already know a neat (a_i, b_i)-embedding of G_i, and we may put $\ell_i = a_i$. If, on the other hand, μ_i is an S-node, then let m be the number of edges in the path skel(μ_i), and let $G_i^1, G_i^2, \ldots, G_i^m$ be the expansion graphs of the edges of the path. For each G_i^j, we have already computed a neat (a_j, b_j)-embedding, so we may now put $\ell_i = \sum_{j=1}^{m} a_j$. Clearly, this value of ℓ_i corresponds to the definition given above.

We now fix two distinct indices $\alpha, \beta \in \{1, \ldots, k\}$, so that the values ℓ_α and ℓ_β are as small as possible; formally, $\ell_\alpha = \min\{\ell_i;\ i = 1, \ldots, k\}$ and $\ell_\beta = \min\{\ell_i;\ i = 1, \ldots, k \text{ and } i \neq \alpha\}$.

Let us fix an embedding of skel(μ) in which e_α and e_β are adjacent to the outer faces. We extend this embedding of skel(μ) into an embedding of pert(μ) by replacing each edge of skel$^*(\mu)$ by a neat embedding of its expansion graph, in such a way that the two boundary paths have lengths ℓ_α and ℓ_β. Let $\mathcal{E}$ be the resulting $(\ell_\alpha, \ell_\beta)$-embedding of pert($\mu$). The embedding $\mathcal{E}$ is neat (we omit the proof).

R-nodes. Suppose now that μ is an R-node. As with P-nodes, we define the *expanded skeleton* skel$^*(\mu)$ by replacing each edge of skel(μ) corresponding to an S-node by a path of appropriate length. The graph skel$^*(\mu)$ together with the parent edge forms a subdivision of a 3-connected graph. In particular, its embedding is determined uniquely up to a flip and a choice of outer face. Fix an embedding of skel$^*(\mu)$ and the parent edge, so that the parent edge is on the outer face. Let f_1 and f_2 be the two faces incident to the parent edge of μ.

Let e be an edge of skel$^*(\mu)$, let G_e be its expansion graph, and let $\mathcal{E}_e$ be a neat (a, b)-embedding of G_e, for some $a \leq b$. The boundary path of $\mathcal{E}_e$ of length a will be called *the short side* of $\mathcal{E}_e$, while the boundary path of length b will be *the long side*. If $a = b$, we choose the long side and short side arbitrarily.

Our goal is to extend the embedding of skel$^*(\mu)$ into an embedding of pert(μ) by replacing each edge e of skel$^*(\mu)$ with a copy of $\mathcal{E}_e$. In doing so, we have to choose which of the two faces incident to e will be adjacent to the short side of $\mathcal{E}_e$.

First of all, if e is an edge of skel$^*(\mu)$ incident to one of the outer faces f_1 or f_2, we embed $\mathcal{E}_e$ in such a way that its short side is adjacent to the outer face. Since f_1 and f_2 do not share an edge in skel$^*(\mu)$, such an embedding is always possible, and guarantees that the resulting embedding of pert(μ) will be out-minimal.

It remains to determine the orientation of $\mathcal{E}_e$ for the edges e that are not incident to the outer faces, in such a way that the largest face of the resulting embedding will be as small as possible. Rather than solving this task optimally, we formulate a linear programming relaxation, and then apply a rounding step which will guarantee a constant factor approximation.

Intuitively, the linear program works as follows: given an edge e incident to a pair of faces f and g, and a corresponding graph G_e with a short side of length a and a long side of length b, rather than assigning the short side to one face and the long side to the other, we assign to each of the two faces a fractional value in the interval $[a, b]$, so that the two values assigned by e to f and g have sum $a + b$, and the maximum total amount assigned to a single face of skel$^*(\mu)$ from its incident edges is as small as possible.

More precisely, we consider the linear program with the set of variables

$$\{M\} \cup \{x_{e,f};\ e \text{ is an edge adjacent to face } f\},$$

where the goal is to minimize M subject to the following constraints:

- For every edge e adjacent to a pair of faces f and g, we have the constraints $x_{e,f} + x_{e,g} = a + b$, $a \leq x_{e,f} \leq b$ and $a \leq x_{e,g} \leq b$, where $a \leq b$ are the lengths of the two boundary paths of $\mathcal{E}_e$.

– Moreover, if an edge e is adjacent to an outer face $f \in \{f_1, f_2\}$ as well as an inner face g, then we set $x_{e,f} = a$ and $x_{e,g} = b$, with a and b as above.
– For every inner face f of $\mathrm{skel}^*(\mu)$, we have the constraint $\sum_e x_{e,f} \leq M$, where the sum is over all edges incident to f.

Given an optimal solution of the above linear program, we determine the embedding of $\mathrm{pert}(\mu)$ as follows: for an edge e of $\mathrm{skel}^*(\mu)$ incident to two inner faces f and g, if $x_{e,f} \leq x_{e,g}$, embed $\mathcal{E}_e$ with its short side incident to f and long side incident to g. Let $\mathcal{E}_\mu$ be the resulting embedding. It can be shown that $\mathcal{E}_\mu$ is neat.

Proposition 1 yields a 6-approximation algorithm for the minimization of largest face in biconnected graphs.

Theorem 4. *A 6-approximation for* MinMaxFace *in biconnected graphs can be computed in polynomial time.*

4 Perfectly Uniform Face Sizes

In this section we study the problem of deciding whether a biconnected planar graph admits a k-uniform embedding. Note that, due to Euler's formula, a connected planar graph with n vertices and m edges has $f = m - n + 2$ faces. In order to admit an embedding where every face has size k, it is necessary that $2m = fk$. Hence there is at most one value of k for which the graph may admit a k-uniform embedding.

In the following, we characterize the graphs admitting 3-uniform and 4-uniform embeddings, and we give an efficient algorithm for testing whether a graph admits a 6-uniform embedding. Finally, we show that testing whether a graph admits a k-uniform embedding is NP-complete for odd $k \geq 7$ and even $k \geq 10$. We leave open the cases $k = 5$ and $k = 8$.

Our characterizations and our testing algorithm use the recursive structure of the SPQR-tree. To this end, it is necessary to consider embeddings of pertinent graphs, where we only require that the interior faces have size k, whereas the outer face may have different size, although it must not be too large. We call such an embedding *almost k-uniform*. The following lemma states that the size of the outer face in such an embedding depends only on the number of vertices and edges in the pertinent graph.

Lemma 1. *Let G be a graph with n vertices and m edges with an almost k-uniform embedding. Then the outer face has length $\ell = k(n - m - 1) + 2m$.*

Thus, for small values of k, where the two boundary paths of the pertinent graph may have only few different lengths, the type of an almost k-uniform embedding (as defined in Section 4) is essentially fixed. Using this fact, the biconnected graphs (with possible multiple edges) admitting a k-uniform dual for $k = 3, 4$ can be characterized. We remark that the simple graphs admitting 3-uniform and 4-uniform embeddings are precisely the maximal planar graphs and the maximal planar bipartite graphs.

Theorem 5. *A biconnected planar graph G admits 3-uniform embedding if and only if its SPQR-tree satisfies all of the following conditions.*

(i) S- and R-nodes are only adjacent to Q- and P-nodes.
(ii) Every R-node skeleton is a planar triangulation.
(iii) Every S-node skeleton has size 3.
(iv) Every P-node with k neighbors has k even and precisely $k/2$ of the neighbors are Q-nodes.

Theorem 6. *A biconnected planar graph admits a 4-regular dual if and only if it is bipartite and satisfies the following conditions.*

(i) For each P-node either all expansion graphs satisfy $m_e = 2n_e - 4$, or half of them satisfy $m_e = 2n_e - 5$ and the other half are Q-nodes.
(ii) For each S- or R-node all faces have size 3 or 4; the expansion graphs of all edges incident to faces of size 4 satisfy $m_e = 2n_e - 3$ and for each triangular face, there is precisely one edge whose expansion graph satisfies $m_e = 2n_e - 4$, the others satisfy $m_e = 2n_e - 3$.

Theorems 5 and 6 can be used to construct linear-time algorithms to test whether a biconnected planar graph admits a 3-regular dual and a 4-regular dual, respectively.

Theorem 7. *It can be tested in $O(n^{1.5})$ time whether a biconnected planar graph admits a 6-uniform embedding.*

Sketch of Proof. To test the existence of a 6-uniform embedding, we again use bottom-up traversal of the SPQR-tree and are therefore interested in the types of almost 6-uniform embeddings of pertinent graphs. Clearly, each of the two boundary paths of a pertinent graph may have length at most 5. Thus, only embeddings of type (a, b) with $1 \leq a \leq b \leq 5$ are relevant. By Lemma 1 the value of $a + b$ is fixed and in order to admit a k-uniform embedding with k even, it is necessary that the graph is bipartite. Thus, for an almost 6-uniform embedding the length of the outer face must be in $\{2, 4, 6, 8, 10\}$. Moreover, the color classes of the poles in the bipartite graph determine the parity of a and b.

For length 2 and length 10, the types must be $(1, 1)$ and $(5, 5)$, respectively. For length 4, the type must be $(1, 3)$ or $(2, 2)$, depending on the color classes of the poles. For length 6, the possible types are $(2, 4)$ or $(3, 3)$ (it can be argued that $(1, 5)$ is not possible). Finally, for length 8, the possible types are $(3, 5)$ and $(4, 4)$ and again the color classes uniquely determine the type.

Thus, we know for each internal node μ precisely what must be the type of an almost 6-uniform embedding of pert(μ) if one exists. It remains to check whether for each node μ, assuming that all children admit an almost 6-uniform embedding, it is possible to put them together to an almost 6-uniform embedding of pert(μ). For this, we need to decide (i) an embedding of skel(μ) and (ii) for each child the flip of its almost k-uniform embedding. The main issue are R-nodes, where we have to solve a generalized matching problem to ensure that every face gets assigned a total boundary length of 6. This can be solved in $O(n^{1.5})$ time [9]. □

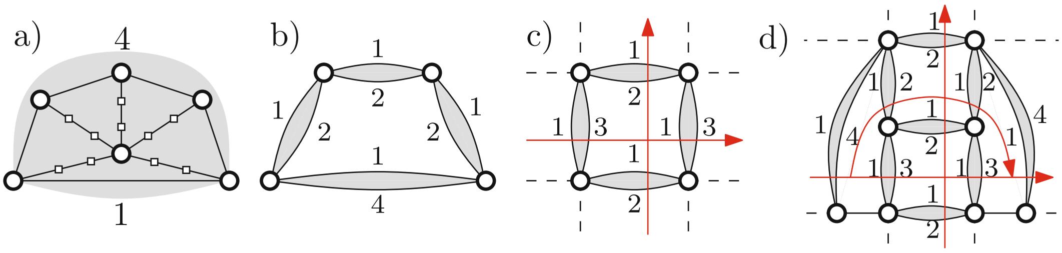


Fig. 2. Illustration of the gadgets used for the hardness proof in Theorem 8. (a) A $(1,4)$-edge. (b) A variable gadget for a variable with three occurrences. (c), (d) Crossing gadgets for a pipe of $(1,2)$-edges with a pipe of $(1,3)$- and $(1,4)$-edges, respectively. The red arrows indicate the information flow.

Finally, we prove NP-hardness for testing the existence of a k-uniform embedding for $k = 7$ and $k \geq 9$ by giving a reduction from the NP-complete problem Planar Positive 1-in-3-SAT where each variable occurs at least twice and at most three times and each clause has size two or three. The NP-completeness of this version of satisfiability follows from the results of Moore and Robson [13].

Theorem 8. *k-UniformFaces is NP-complete for all odd $k \geq 7$ and even $k \geq 10$.*

Sketch of Proof. We sketch the reduction for $k = 7$, the other cases are similar. We reduce from Planar Positive 1-in-3-SAT where each variable occurs two or three times and each clause has size two or three. Essentially, we perform a standard reduction, replacing each variable, each clause, and each edge of the variable–clause graph by a corresponding gadget, similar to the proof of Theorem 1.

First, it is possible to construct subgraphs that behave like $(1,2)$-, $(1,3)$-, and $(1,4)$-edges, i.e., their embedding is unique up to a flip, the inner faces have size 7 and the outer face has a path of length 1 and a path of length 2, 3 or 4 between the poles; see Fig. 2a for an example.

A variable is a cycle consisting of $(1,2)$-edges, called *output edges* (one for each occurrence of the variable) and one *sink-edge*, which is a $(1,3)$- or a $(1,4)$-edges depending on whether the variable occurs two or three times; see Fig. 2b. It is not hard to construct clauses with two or three incident $(1,2)$-edges whose internal face has size 7 if and only if exactly one of the incident $(1,2)$-edges contributes a path of length 1 to the internal face. We then use simple pipes to transmit the information encoded in the output edges to the clauses in a planar way. The main issue are the sink-edges, which have different length depending on whether the corresponding variable is `true` or `false`. To this end, we transfer the information encoded in all sink edges via pipes to a single face. We use the fact that sink-edges are $(1,k)$-edges with $k > 2$ to cross over pipes transmitting these values using the crossing gadgets shown in Fig. 2c,d. Note that the construction in Fig. 2d is necessary since crossing a pipe of $(1,4)$-edges with a pipe of $(1,2)$-edges in the style of Fig. 2c would require face size at least 8.

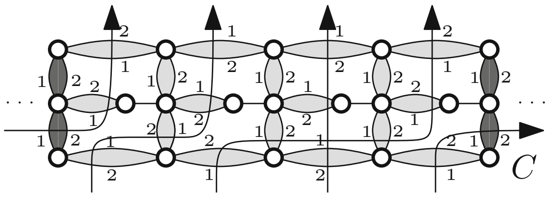


Fig. 3. Illustration of a shift ring (the left and right dark gray edges are identified) that allows to transpose adjacent $(1, 2)$-edges encoding different states. The read arrows show the flow of information encoded in the $(1, 2)$-edges.

Now we have collected all the information encoded in the sink edges in a single face. By attaching variable gadgets to each of the corresponding pipes, we split this information into $(1, 2)$-edges, whose endpoints we identify such that they all form a single large cycle C.

Now, for all faces except for the inner faces of the gadgets and the face inside cycle C, we apply the following simple procedure. We triangulate them and insert into each triangle a new vertex connected to all its vertices by edges subdivided sufficiently often so that all faces have size 7. As a result the only remaining embedding choices are the flips of the $(1, d)$-edges used in the gadgets. We have that the original 1-in-3SAT formula is satisfiable if and only if the $(1, d)$-edges can be flipped so that all faces except the one inside C have size 7.

It follows from Lemma 1 that the length of the face inside C is uniquely determined if all other faces have size 7, but we do not know which of the $(1, 2)$-edges contribute paths of length 1 and which contribute paths of length 2. It then remains to give a construction that can subdivide the interior of C into faces of size 7 for any possible distribution. This is achieved by inserting ring-like structures that allow to shift and transpose adjacent edges that contribute paths of length 1 and length 2; see Fig. 3. By nesting sufficiently many such rings, we can ensure that in the innermost face the edges contributing paths of length 1 are consecutive, and the first one (in some orientation of C) is at a fixed position. Then we can assume that we know exactly what the inner face looks like and we can use one of the previous constructions to subdivide it into faces of size 7. □

5 Conclusions and Open Problems

Throughout the paper we consider embeddings of planar multi-graphs on the sphere, that is, no face is regarded as the outer face. On the other hand, when dealing with embeddings in the plane, it seems natural to consider the variants of the MinMaxFace and UniformFaces problems in which the size of the outer face is not constrained. To simplify the analysis, we decided not to explicitly discuss these variants. However, both the hardness results and the embedding algorithms presented in the paper can be easily modified to handle outer faces of arbitrary size. In fact, in the reduction for k-MinMaxFace and k-UniformFaces, we can insert in any triangular face a cycle of arbitrary length and triangulate any but the inner face bounded by the cycle. Also, when computing the type of an embedding of a node of the SPQR-tree, it is not difficult to additionally consider embeddings of the corresponding pertinent graph in which one of the two paths bounding the outer face has length greater than or equal to k.

We list some interesting open questions: What is the complexity of k-UniformFaces for $k = 5$ and $k = 8$? Are UniformFaces and MinMaxFace polynomial-time solvable for biconnected series-parallel graphs? Are they FPT with respect to treewidth?

Acknowledgments. We thank Bartosz Walczak for discussions.

References

1. Angelini, P., Di Battista, G., Patrignani, M.: Finding a minimum-depth embedding of a planar graph in $O(n^4)$ time. Algorithmica **60**, 890–937 (2011)
2. Bienstock, D., Monma, C.L.: On the complexity of covering vertices by faces in a planar graph. SIAM J. Comput. **17**(1), 53–76 (1988)
3. Bläsius, T., Krug, M., Rutter, I., Wagner, D.: Orthogonal graph drawing with flexibility constraints. Algorithmica **68**, 859–885 (2014)
4. Bläsius, T., Rutter, I., Wagner, D.: Optimal orthogonal graph drawing with convex bend costs. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) ICALP 2013, Part I. LNCS, vol. 7965, pp. 184–195. Springer, Heidelberg (2013)
5. Da Lozzo, G., Jelínek, V., Kratochvíl, J., Rutter, I.: Planar Embeddings with Small and Uniform Faces. ArXiv e-prints (September 2014)
6. Di Battista, G., Liotta, G., Vargiu, F.: Spirality and optimal orthogonal drawings. SIAM Journal on Computing **27**(6), 1764–1811 (1998)
7. Di Battista, G., Tamassia, R.: On-line graph algorithms with SPQR-trees. In: Paterson, M. (ed.) ICALP 1990. LNCS, vol. 443, pp. 598–611. Springer, Heidelberg (1990)
8. Fellows, M.R., Kratochvíl, J., Middendorf, M., Pfeiffer, F.: The complexity of induced minors and related problems. Algorithmica **13**, 266–282 (1995)
9. Gabow, H.N.: An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In: Theory of Computing, STOC 1983, pp. 448–456. ACM (1983)
10. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. SIAM J. on Comput. **31**(2), 601–625 (2001)
11. Gutwenger, C., Mutzel, P.: A linear time implementation of SPQR-trees. In: Marks, J. (ed.) GD 2000. LNCS, vol. 1984, pp. 77–90. Springer, Heidelberg (2001)
12. Gutwenger, C., Mutzel, P.: Graph embedding with minimum depth and maximum external face. In: Liotta, G. (ed.) GD 2003. LNCS, vol. 2912, pp. 259–272. Springer, Heidelberg (2004)
13. Moore, C., Robson, J.M.: Hard tiling problems with simple tiles. Discrete Comput. Geom. **26**(4), 573–590 (2001)
14. Mutzel, P., Weiskircher, R.: Optimizing over all combinatorial embeddings of a planar graph. In: Cornuéjols, G., Burkard, R.E., Woeginger, G.J. (eds.) IPCO 1999. LNCS, vol. 1610, pp. 361–376. Springer, Heidelberg (1999)
15. Woeginger, G.J.: Embeddings of planar graphs that minimize the number of long-face cycles. Oper. Res. Lett., 167–168 (2002)

Appendix H

Planarity of Partially Embedded Graphs

Testing Planarity of Partially Embedded Graphs

PATRIZIO ANGELINI and GIUSEPPE DI BATTISTA, Roma Tre University, Italy
FABRIZIO FRATI, University of Sydney, Australia
VÍT JELÍNEK and JAN KRATOCHVÍL, Charles University, Prague, Czech Republic
MAURIZIO PATRIGNANI, Roma Tre University, Italy
IGNAZ RUTTER, Karlsruhe Institute of Technology (KIT), Germany and Charles University, Prague

We study the following problem: given a planar graph G and a planar drawing (embedding) of a subgraph of G, can such a drawing be extended to a planar drawing of the entire graph G? This problem fits the paradigm of extending a partial solution for a problem to a complete one, which has been studied before in many different settings. Unlike many cases, in which the presence of a partial solution in the input makes an otherwise easy problem hard, we show that the planarity question remains polynomial-time solvable. Our algorithm is based on several combinatorial lemmas, which show that the planarity of partially embedded graphs exhibits the 'TONCAS' behavior "the obvious necessary conditions for planarity are also sufficient." These conditions are expressed in terms of the interplay between (1) the rotation system and containment relationships between cycles and (2) the decomposition of a graph into its connected, biconnected, and triconnected components. This implies that no dynamic programming is needed for a decision algorithm and that the elements of the decomposition can be processed independently.

Further, by equipping the components of the decomposition with suitable data structures and by carefully splitting the problem into simpler subproblems, we make our algorithm run in linear time.

Finally, we consider several generalizations of the problem, such as minimizing the number of edges of the partial embedding that need to be rerouted to extend it, and argue that they are NP-hard. We also apply our algorithm to the simultaneous graph drawing problem SIMULTANEOUS EMBEDDING WITH FIXED EDGES (SEFE). There we obtain a linear-time algorithm for the case that one of the input graphs or the common graph has a fixed planar embedding.

Categories and Subject Descriptors: G.2.2 [**Algorithms**]

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Planarity, partial embedding, simultaneous embedding, algorithm

A preliminary version of this article appeared as "Testing Planarity of Partially Embedded Graphs" in *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, pages 202–221.

Work on the journal version of this article was supported by ESF EuroGIGA GraDR as Czech Research grant GACR GIG-11-E023 for V. Jelínek, J. Kratochvíl, and I. Rutter (partial support). I. Rutter was supported by a fellowship within the postdoctoral program of the German Academic Exchange Service (DAAD). F. Frati acknowledges support from the Australian Research Council (grant DE140100708). Research was supported in part by the MIUR project AMANDA (Algorithmics for Massive and Networked Data) protocol 2012C4E3KT_001.

Authors' addresses: P. Angelini, G. Di Battista, and M. Patrignani, Dipartimento di Ingegneria, Universita' Roma Tre, Via della Vasca Navale 79, 00146, Rome, Italy; emails: {angelini, gdb, patrigna}@dia.uniroma3.it; F. Frati, School of Information Technologies, The University of Sydney, NSW 2006, Sydney, Australia; email: fabrizio.frati@sydney.edu.au; V. Jelínek, IUUK MFF UK, Malostranske nam 25, 11800 Praha 1, Czech; email: jelinek@iuuk.mff.cuni.cz; J. Kratochvíl, KAM MFF UK, Malostranske nam 25, 11800 Praha 1, Czech Republic; email: honza@kam.mff.cuni.cz; I. Rutter, Karlsruhe Institute of Technology (KIT), Institute of Theoretical Informatics, Box 6980, D-76128, Karlsruhe, Germany; email: rutter@kit.edu.

DOI: http://dx.doi.org/10.1145/2629341

ACM Reference Format:
Patrizio Angelini, Giuseppe Di Battista, Fabrizio Frati, Vít Jelínek, Jan Kratochvíl, Maurizio Patrignani, and Ignaz Rutter. 2015. Testing planarity of partially embedded graphs. ACM Trans. Algor. 11, 4, Article 32 (April 2015), 42 pages.
DOI: http://dx.doi.org/10.1145/2629341

1. INTRODUCTION

Planarity is one of the central concepts not only in graph drawing but in graph theory as a whole. The characterization of planar graphs proved by Kuratowski [1930] represents a fundamental result in modern graph theory. This characterization, based on two forbidden topological subgraphs—K_5 and $K_{3,3}$—makes planarity a finite problem and leads to a polynomial-time recognition algorithm. Planarity is thus "simple" from the computational point of view (this, of course, does not mean that algorithms for testing planarity are trivial) in the strongest possible way, as several linear-time algorithms for testing planarity are known [Boyer and Myrvold 2004; de Fraysseix et al. 2006; Hopcroft and Tarjan 1974].

In this article, we pose and study the question of planarity testing in a constrained setting, namely when part of the input graph is already drawn and cannot be changed. Practical motivation for this question comes from the visualization of large networks in which certain patterns are required to be drawn in a standard way. The known planarity testing algorithms, even those that build a drawing incrementally, are of no help here, as they are allowed to redraw at each step the part of the graph processed so far. For similar reasons, online planar embedding and planarity testing algorithms, such as those of Di Battista and Tamassia [1996], Poutré [1994], Tamassia [1996], and Westbrook [1992], are not suitable to be used in this context.

Related work. The question of testing the planarity of partially drawn graphs fits into the general paradigm of extending a partial solution for a problem to a full one. This has been studied in various settings, and often the extendability problem is more difficult than the unconstrained one. As an example, graph coloring is NP-complete for perfect graphs even if only four vertices are already colored [Kratochvíl and Sebo 1997], whereas the chromatic number of a perfect graph can be determined in polynomial time [Grötschel et al. 1988]. Another example is provided by edge colorings—deciding 3-edge-colorability of cubic bipartite graphs if some edges are already colored is NP-complete [Fiala 2003], whereas it follows from the famous Kőnig-Hall theorem that cubic bipartite graphs are always 3-edge colorable. In view of these hardness results, it is somewhat surprising that the planarity of partially drawn graphs can be tested in polynomial time, in fact in linear time, as we show in this article. This is all the more so, considering that this problem is known to be NP-hard for drawings where edges are constrained to be straight-line segments [Patrignani 2006].

Specific constraints on planar graph drawings have been studied by several authors (e.g., see Dornheim [2002], Gutwenger et al. [2008], Tamassia [1998], and Tamassia et al. [1988]). However, none of those results can be exploited to solve the question that we pose in this article. The work in Juvan and Mohar [2005] and Mohar [1999] give algorithms for extending 2-cell embeddings on the torus and surfaces of higher genus. Their results show that even for arbitrary surfaces, the problem of extending an embedding of a graph $H \subseteq G$ to an embedding of G is fixed-parameter tractable with respect to the branch size of H. The *branch size* of a graph H is the size of the smallest graph H' from which H can be obtained as a subdivision. However, the approach of Juvan and Mohar is not applicable to our problem, as our goal is to find algorithms that are polynomial in the size of H as well as G. Moreover, the algorithm by Juvan and Mohar assumes that either each component of $G - H$ has at most two allowed embeddings, which are given as part of the input, or H has a *closed* 2-cell embedding—that is, H is biconnected.

Contribution and outline. To solve the general problem, we allow disconnected graphs or graphs with low connectivity to be part of the input. It is readily seen that in this case the rotation system (i.e., the cyclic orderings of the edges incident to the vertices of the graph) does not fully describe the input. In fact, the relative position of vertices against cycles in the graph must also be considered. (These concepts and their technical details are discussed later.) Further, we make use of the fact that drawing graphs on the plane and on the sphere are equivalent concepts. The advantage of considering embeddings on the sphere lies in the fact that we do not need to distinguish between the outer face and the inner faces.

Many known planarity testing algorithms work by incrementally extending a partial drawing constructed in previous steps. The main idea of our algorithm is to look at the problem from the "opposite" perspective. Namely, we do not try to directly extend the input partial embedding (which seems much harder than one would expect). Instead, we look at the possible embeddings of the entire graph and decide if any of them contains the embedding of the subgraph prescribed by the input.

Our algorithm is based on several combinatorial lemmas, relating the problem to the connectivity of the graph. Most of them exhibit the TONCAS property—"the obvious necessary conditions are also sufficient." This is particularly elegant in the case of 2-connected graphs, in which we exploit the SPQR-tree decomposition of the graph. This notion was introduced by Di Battista and Tamassia [1996] to describe all possible embeddings of 2-connected planar graphs in a succinct way and has been used in various situations when asking for planar embeddings with special properties. A survey on the use of this technique in planar graphs is given by Mutzel [2003]. It is indeed obvious that if a 2-connected graph admits an embedding extending a given partial embedding, then the skeleton of each node of the SPQR-tree has a drawing *compatible* (a precise definition of compatibility will come later) with the partial embedding. We prove that the converse is also true. Hence, if we only aim at polynomial running time, we do not need to perform *any* dynamic programming on the SPQR-tree and could process its nodes independently. However, for the ultimate goal of linear running time, we must refine the approach and pass several pieces of information through the SPQR-tree. Then, dynamic programming becomes very useful. In addition, the SPQR-trees are exploited at two levels of abstraction, both for decomposing an entire block and for computing the embedding of the subgraph induced by each face of the constrained part of the drawing.

The article is organized as follows. We first describe the terminology and list auxiliary topological lemmas in Section 2. In particular, the combinatorial invariants of equivalent embeddings are introduced. In Section 3, we state the combinatorial characterization theorems for 2-connected, connected, and disconnected cases. These theorems yield a simple polynomial-time algorithm outlined at the end of the section. Section 4 is devoted to the technical details of the linear-time algorithm. Section 5 discusses several possible generalizations of the partially embedded planarity concept leading to NP-hard problems and shows how our techniques can be used to solve other graph drawing problems. We summarize our results and discuss some directions for further research in Section 6.

2. NOTATION AND PRELIMINARIES

In this section, we introduce some notations and preliminaries that we use throughout the article. In particular, we give a detailed description of how planar embeddings of

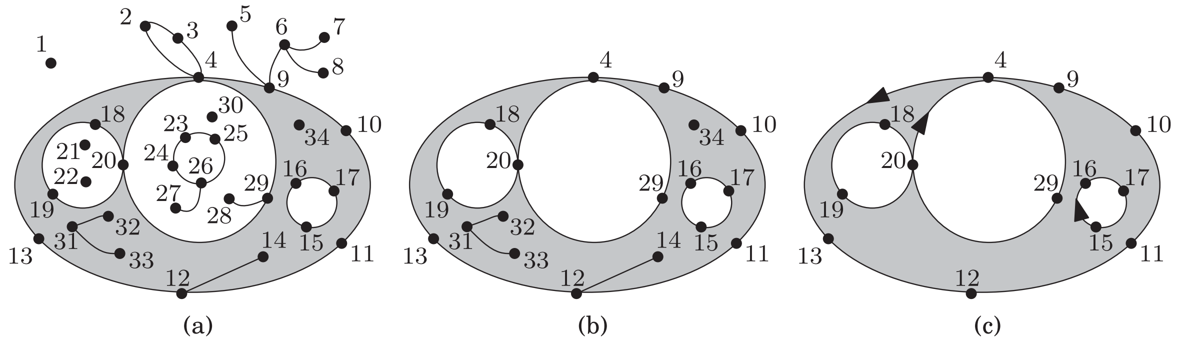


Fig. 1. (a) A planar drawing of a graph G. The shaded region represents a face f of the drawing. (b) The boundary of f. The circular lists defining the boundary of f are [15, 16, 17], [33, 31, 32, 31], [13, 12, 14, 12, 11, 10, 9, 4, 29, 20, 19, 18, 20, 4], [34]. (c) The facial cycles of f.

not necessarily connected graphs can be handled, and we give a first characterization of the embeddings extending given partial embeddings. We conclude with an overview of data structures and their efficient construction, which will be particularly important for the linear-time implementation of our algorithm.

The definitions listed in this section are standard and can be found in most graph theory textbooks. We are listing them for the sake of completeness. Perhaps less standard is the notion of H-bridges (first introduced by Demoucron et al. [1964] under the name *fragment*) and the definition of SPQR-trees (introduced in Di Battista and Tamassia [1996], we have followed an alternative wording used in the conference version of this paper [Angelini et al. 2010]).

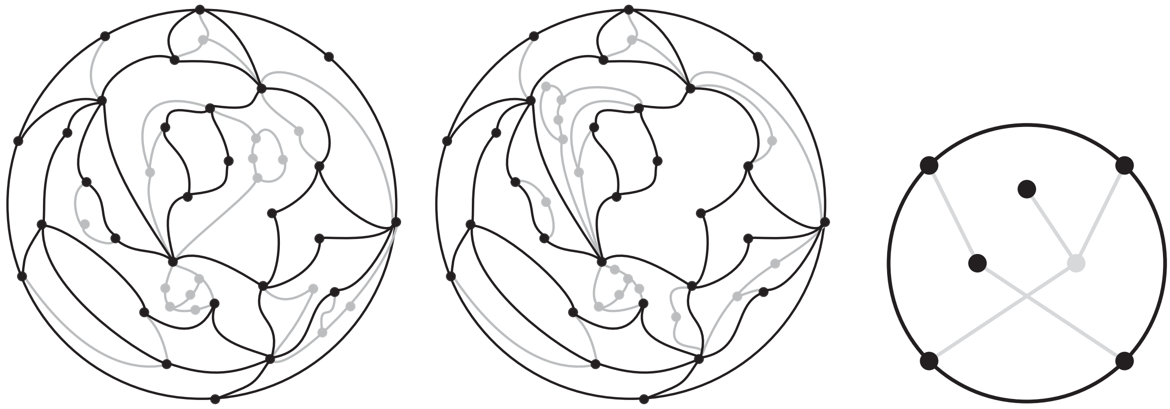
2.1. Drawings, Embeddings, and the Problem Definition

A *drawing* of a graph is a mapping of each vertex to a distinct point of the plane and of each edge to a simple curve connecting its endpoints. A drawing is *planar* if the curves representing its edges do not intersect, except, possibly, at common endpoints. A graph is *planar* if it admits a planar drawing. A planar drawing Γ determines a subdivision of the plane into connected regions, called *faces*, and a circular ordering of the edges incident to each vertex, called *rotation system*. The circular ordering of the edges incident to a vertex x is the *(local) rotation of* x.

Visiting the (not necessarily connected) border of a face f of Γ in such a way to keep f to the left, we determine a set of circular lists of vertices. Such a set is the *boundary* of f. Two drawings are *equivalent* if they have the same rotation system and the same face boundaries. A *planar embedding* is an equivalence class of planar drawings. Note that equivalent planar drawings need not have the same outer face, and that a planar embedding does not determine which face is the outer face. This loss of information is harmless, as for the purposes of extending a partial embedding, the choice of the outer face is irrelevant. It is therefore convenient to imagine planar graphs as being embedded on a sphere, where no face plays the special role of outer face.

For connected graphs in the plane, an embedding is uniquely determined by the rotation system. For disconnected graphs, on the other hand, this information is not sufficient, as it does not determine the relative positions of the connected components. However, this additional information is encoded in the face boundaries, which, together with the rotation system, completely describe planar embeddings, even for disconnected graphs; Figure 1(a) and (b) provide an example.

Our initial motivation was to extend a given drawing of a subgraph of a planar graph to a planar drawing of the entire graph; however, it is not hard to see that this is equivalent to an embedding problem, where we wish to extend a planar embedding of a subgraph to a planar embedding of the whole graph.



(a) (b) (c)

Fig. 2. Two different planar embeddings of a graph G whose restrictions to H (black vertices and edges) coincide with $\mathcal{H}$ (a, b). An instance that does not admit an embedding extension (c). Vertices and edges in $G\backslash H$ are grey.

A *partially embedded graph*, or PEG for short, is a triplet $(G, H, \mathcal{H})$, where G is a graph, H is a subgraph of G, and $\mathcal{H}$ is a planar embedding of H. We say that the vertices and edges of H are *prescribed*. The problem PARTIALLY EMBEDDED PLANARITY (PEP) asks whether a given PEG $(G, H, \mathcal{H})$ admits a planar embedding $\mathcal{G}$ of G whose restriction to H is $\mathcal{H}$. In this case, we say that the PEG $(G, H, \mathcal{H})$ is *planar*. We say that $\mathcal{G}$ is an *extension* of an embedding $\mathcal{H}$ of H if the restriction of $\mathcal{G}$ to H is $\mathcal{H}$. Figure 2 presents an example of a PEG that admits several different embedding extensions and an example that does not admit any.

2.2. Facial Cycles

Let Γ be a planar drawing of a graph H (see Figure 1(a)). Let $\vec{C}$ be a simple cycle in H with an arbitrary orientation. The oriented cycle $\vec{C}$ splits the plane into two connected parts. Denote by $V_\Gamma^{\text{left}}(\vec{C})$ and $V_\Gamma^{\text{right}}(\vec{C})$ the sets of vertices of the graph that are to the left and to the right of $\vec{C}$ in Γ, respectively. The boundary of each face f of Γ can be uniquely decomposed into simple edge-disjoint cycles, *bridges* (edges that are not part of any cycle), and isolated vertices see Figure 1(b)). Orient the cycles in such a way that f is to the left when walking along the cycle according to the orientation. Call these oriented cycles the *facial cycles* of f (see Figure 1(c)). Observe that the sets $V_\Gamma^{\text{left}}(\vec{C})$, $V_\Gamma^{\text{right}}(\vec{C})$, and the notion of facial cycles only depend on the embedding $\mathcal{H}$ of Γ. Hence, it makes sense to write $V_\mathcal{H}^{\text{left}}(\vec{C})$ and $V_\mathcal{H}^{\text{right}}(\vec{C})$, and to define the facial cycles of $\mathcal{H}$ as the facial cycles of the faces of $\mathcal{H}$.

For a vertex x of a graph G with embedding $\mathcal{G}$, we denote by $E_G(x)$ the set of edges incident to x and by $\sigma_\mathcal{G}(x)$ the (local) rotation of x in $\mathcal{G}$. The following lemma characterizes the planar embeddings of a PEG $(G, H, \mathcal{H})$ that extend $\mathcal{H}$ in terms of the rotation system and relative cycle–vertex positions with respect to the facial cycles of $\mathcal{H}$.

LEMMA 2.1. *Let $(G, H, \mathcal{H})$ be a PEG and let $\mathcal{G}$ be a planar embedding of G. The restriction of $\mathcal{G}$ to H is $\mathcal{H}$ if and only if the following conditions hold:*

(1) *for every vertex $x \in V(H)$, $\sigma_\mathcal{G}(x)$ restricted to $E_H(x)$ coincides with $\sigma_\mathcal{H}(x)$, and*
(2) *for every facial cycle $\vec{C}$ of each face of $\mathcal{H}$, we have that $V_\mathcal{H}^{\text{left}}(\vec{C}) \subseteq V_\mathcal{G}^{\text{left}}(\vec{C})$ and $V_\mathcal{H}^{\text{right}}(\vec{C}) \subseteq V_\mathcal{G}^{\text{right}}(\vec{C})$.*

PROOF. The proof follows easily from the following statement. Let Γ_1 and Γ_2 be two drawings of the same graph G such that for every vertex $x \in V(G)$, $\sigma_{\Gamma_1}(x) = \sigma_{\Gamma_2}(x)$ holds. Assume that each facial cycle $\vec{C}$ for any face f in Γ_1 or in Γ_2 is oriented in such a way that f is to the left of $\vec{C}$. Drawings Γ_1 and Γ_2 are equivalent if and only if they have the same oriented facial cycles and, for each oriented facial cycle $\vec{C}$, it holds $V^{\text{left}}_{\Gamma_1}(\vec{C}) = V^{\text{left}}_{\Gamma_2}(\vec{C})$.

We need to prove this statement in both directions: (i) if Γ_1 and Γ_2 have the same embedding, then they have the same oriented facial cycles and, for each facial cycle $\vec{C}$, we have $V^{\text{left}}_{\Gamma_1}(\vec{C}) = V^{\text{left}}_{\Gamma_2}(\vec{C})$, and (ii) if Γ_1 and Γ_2 have the same oriented facial cycles and, for each facial cycle $\vec{C}$, we have $V^{\text{left}}_{\Gamma_1}(\vec{C}) = V^{\text{left}}_{\Gamma_2}(\vec{C})$, then Γ_1 and Γ_2 have the same embedding.

We start with direction (i). By definition, drawings with the same embedding have the same facial boundaries and hence the same oriented facial cycles. Suppose for a contradiction that for some facial cycle $\vec{C}$, $V^{\text{left}}_{\Gamma_1}(\vec{C}) \neq V^{\text{left}}_{\Gamma_2}(\vec{C})$. Then, at least one vertex v is to the left of $\vec{C}$ in Γ_1 and to the right of $\vec{C}$ in Γ_2 (the opposite case being analogous). Hence, v is part of the boundary of a face that is to the left of $\vec{C}$ in Γ_1 and part of the boundary of a face that is to the right of $\vec{C}$ in Γ_2, contradicting the hypothesis that Γ_1 and Γ_2 have the same facial boundaries.

We now come to the proof of direction (ii). First, suppose that G is connected and has at least one vertex of degree 3. In this case, the fact that Γ_1 and Γ_2 have the same rotation system implies that they also have the same face boundaries and, hence, the same embedding. Second, suppose that G is connected and has maximum degree 2. Then, G is either a path or a cycle. In both cases, the face boundaries of Γ_1 and Γ_2 are the same (recall that G is drawn on the sphere). Finally, suppose that G has several connected components $C_1, C_2, \ldots, C_k$. We say that two components C_i and C_j *share a face* f if there exists a vertex of C_i and a vertex of C_j on the boundary of f. The drawings Γ_1 and Γ_2 have the same face boundaries if (a) for each C_i, with $i = 1, \ldots, k$, the embedding $\mathcal{G}_1$ of G in Γ_1 restricted to C_i is the same as the embedding $\mathcal{G}_2$ of G in Γ_2 restricted to C_i, and (b) each pair of connected components C_i and C_j, with $i, j \in \{1, \ldots, k\}$ and $i \neq j$, either do not share a face both in $\mathcal{G}_1$ and in $\mathcal{G}_2$ or they contribute with the same circular lists to the boundary of the same face f in $\mathcal{G}_1$ and in $\mathcal{G}_2$.

It can be seen that condition (a) is satisfied by applying the argument we made for a connected graph to each connected component of G.

Condition (b) follows from the hypothesis that for each oriented facial cycle $\vec{C}$, we have $V^{\text{left}}_{\Gamma_1}(\vec{C}) = V^{\text{left}}_{\Gamma_2}(\vec{C})$. Suppose, for a contradiction, that two connected components C_x and C_y of G share a face f in $\mathcal{G}_1$ and no face in $\mathcal{G}_2$. Since C_x and C_y share a face in $\mathcal{G}_1$, they are on the same side of any facial cycle $\vec{C}$ belonging to any other component C_z of G (more intuitively, C_x and C_y are not separated by any cycle and in particular by any facial cycle in Γ_1). On the other hand, since C_x and C_y do not share a face in Γ_2, there exists a component C_z of G containing a facial cycle $\vec{C}$ separating C_x from C_y, thus contradicting the hypothesis that $V^{\text{left}}_{\Gamma_1}(\vec{C}) = V^{\text{left}}_{\Gamma_2}(\vec{C})$.

Next suppose, for a contradiction, that two connected components C_x and C_y contribute with circular lists L^x_1 and L^y_1 to the boundary of the same face f_1 of $\mathcal{G}_1$ and with circular lists L^x_2 and L^y_2 to the boundary of the same face f_2 of $\mathcal{G}_2$ and suppose that $L^x_1 \neq L^x_2$. In particular, assume, without loss of generality, that there exists a facial cycle $\vec{C}'$ of f_2 that is part of C_x and that is not a facial cycle of f_1. The boundary of f_1 is

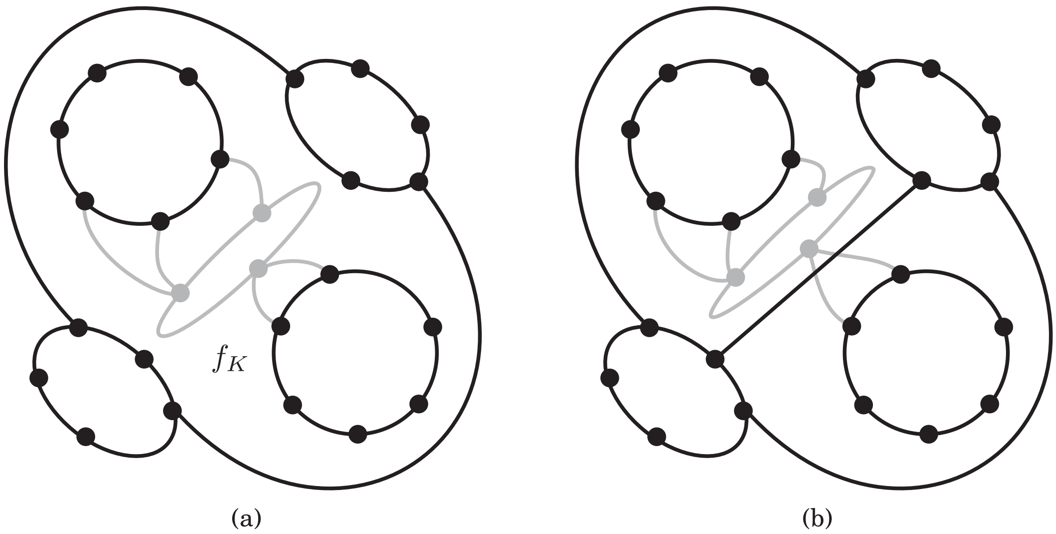


Fig. 3. A nonlocal bridge is either necessarily contained in a face f_K (a) or causes a nonplanarity (b).

oriented in such a way that every facial cycle of f_1 has f_1 to its left. Then, every facial cycle of f_1 obtained from L_1^x has C_y to its left. Further, there exists a facial cycle $\vec{C}$ of f_1 obtained from L_1^x that has $\vec{C}'$ to its right (part of $\vec{C}$ and of $\vec{C}'$ may coincide). As $\mathcal{G}_1$ and $\mathcal{G}_2$ restricted to C_x give the same embedding, the last statement is true both in $\mathcal{G}_1$ and in $\mathcal{G}_2$. Since $\vec{C}'$ is incident to f_2 and since C_y is incident to f_2, such a component is to the right of $\vec{C}$, contradicting the hypothesis that $V^{\text{left}}_{\Gamma_1}(\vec{C}) = V^{\text{left}}_{\Gamma_2}(\vec{C})$. $\square$

2.3. Connectivity, *H*-Bridges, and Data Structures

A graph is *connected* if every pair of vertices is connected by a path. A *k-connected* graph G is such that removing any at most $k-1$ vertices leaves G connected; 3-connected, 2-connected, and 1-connected graphs are also called *triconnected*, *biconnected*, and *simply connected* graphs, respectively. By convention, the complete graph on k vertices is $(k-1)$-connected but not k-connected. A *separating k-set* is a set of k vertices whose removal disconnects the graph. Separating 1- and 2-sets are called *cutvertices* and *separation pairs*, respectively. Hence, a connected graph is biconnected if it has no cutvertices, and it is triconnected if it has no separation pairs and no cutvertices. A *block* of G is either a maximal biconnected subgraph of G or a bridge in G. Each edge of G falls into a single block of G, whereas cutvertices are shared by different blocks. We extend the notion of k-connectivity to PEGs by saying that a PEG $(G, H, \mathcal{H})$ is k-connected if and only if G is k-connected.

Let G be a graph and let H be a subgraph of G. An *H-bridge* K of G is a subgraph of G formed either by a single edge $e \in E(G)\setminus E(H)$ whose end-vertices belong to H or by a connected component K^- of $G - V(H)$, together with all edges (and their end-vertices) that connect a vertex in K^- to vertices in H. In the first case, the H-bridge is *trivial*. A vertex that belongs to $V(H) \cap V(K)$ is called an *attachment vertex* (or *attachment*) of K. Note that the edge sets of the H-bridges form a partition of $E(G)\setminus E(H)$.

An H-bridge K is *local* to a block B of H if all attachments of K belong to B. Notice that an H-bridge with a single attachment can be local to more than one block, whereas an H-bridge with at least two attachments is local to at most one block. An H-bridge that is not local to any block of H is *nonlocal*.

Note that for a nonlocal H-bridge K, there exists at most one face of $\mathcal{H}$ containing all attachments of K (Figure 3(a)). Namely, if all attachments of K were contained on the boundaries of two distinct faces of $\mathcal{H}$, then K would necessarily be local. In addition, if there is no face of $\mathcal{H}$ incident to all attachments of K, then G clearly has no embedding extension (see Figure 3(b)).

Let $(G, H, \mathcal{H})$ be a PEG. In the following, we define some data structures that are widely used throughout the article. All of these data structures can easily be computed in time linear in the number of edges of the graph or of the embedding to which they refer. We use the decomposition of a graph G into its connected, biconnected, and triconnected components. To further relate these decompositions with the embedding of $\mathcal{H}$, we make use of several auxiliary data structures.

The *component–face tree* $\mathcal{CF}$ of $\mathcal{H}$ is a tree whose nodes are the connected components of H and the faces of $\mathcal{H}$. A face f and a component C are joined by an edge if a vertex of C is incident to f. The *block–face tree* $\mathcal{BF}$ of $\mathcal{H}$ is a tree whose nodes are the blocks of H and the faces of $\mathcal{H}$. A face f and a block B are joined by an edge if B contains an edge incident to f. The *vertex–face incidence graph* $\mathcal{VF}$ of $\mathcal{H}$ is a graph whose nodes are the vertices of H and the faces of $\mathcal{H}$. A vertex v and a face f are joined by an edge if v appears on the boundary of f.

To handle the decomposition of a graph into biconnected components, we use the *block-cutvertex tree*. The block-cutvertex tree of a connected graph G is a tree whose nodes are the blocks and the cutvertices of G. Edges in the block-cutvertex tree join each cutvertex to the blocks to which it belongs. The *enriched block-cutvertex tree* of G is a tree obtained by adding to the block-cutvertex tree of G each vertex v of G that is not a cutvertex and by connecting v to the unique block to which it belongs.

To handle the decomposition of a graph into its triconnected components, we use the SPQR-tree $\mathcal{T}$ of G, which we describe in the following.

Let G be a graph. A *split pair* of G is either a separation pair or a pair of adjacent vertices. A *maximal split component*[1] of G with respect to a split pair $\{u, v\}$ (or, simply, a maximal split component of $\{u, v\}$) is either an edge (u, v) or a maximal subgraph G' of G such that G' contains u and v, and $\{u, v\}$ is not a split pair of G'. A vertex $w \neq u, v$ belongs to exactly one maximal split component of $\{u, v\}$. A *split component* of $\{u, v\}$ is the union of any number of maximal split components of $\{u, v\}$.

The SPQR-tree $\mathcal{T}$ of a biconnected graph G is a data structure that describes a recursive decomposition of G induced by its split pairs. The nodes of $\mathcal{T}$ are of four types: S, P, Q, and R. The Q-nodes are the leaves of the tree $\mathcal{T}$. Each Q-node represents a unique edge of the graph G.

Each node μ of $\mathcal{T}$ has an associated biconnected multigraph, called the *skeleton of* μ and denoted by skel(μ). The skeleton describes a decomposition of G into edge-disjoint split components. The edges of the skeleton are called *virtual edges*.

For an internal node μ of $\mathcal{T}$ of degree d, the skeleton skel(μ) has d virtual edges $e_1, e_2, \ldots, e_d$, which correspond bijectively to the connected components of $\mathcal{T} - \mu$. Let $\mathcal{T}_i$ be the component of $\mathcal{T} - \mu$ corresponding to e_i, and let G_i be the subgraph of G formed by all edges of G whose Q-nodes belong to $\mathcal{T}_i$. The graph G_i is called *the expansion graph* of e_i. Each expansion graph G_i is a split component of G, and by replacing each virtual edge e_i in skel(μ) with its expansion graph G_i, we obtain the graph G.

For algorithmic purposes, it is often convenient to treat $\mathcal{T}$ as a rooted tree. In such a case, we choose an arbitrary Q-node as a root of $\mathcal{T}$. In the skeleton of any nonroot node μ, there is a unique virtual edge representing the component of $\mathcal{T} - \mu$ that contains the root. This virtual edge is the *parent edge* of skel(μ). Supposing that μ has k children, let $e_1, \ldots, e_k$ be the nonparent edges of skel(μ), and let $G_1, \ldots, G_k$ be their expansion graphs. The graph $G^* = G_1 \cup G_2 \cup \cdots \cup G_k$ is the *pertinent graph of* μ, denoted by pert(μ).

[1]Note that "maximal" refers to the splitting (the component cannot be split further by $\{u, v\}$), not the size, of the component. We use the term for consistency with existing literature.

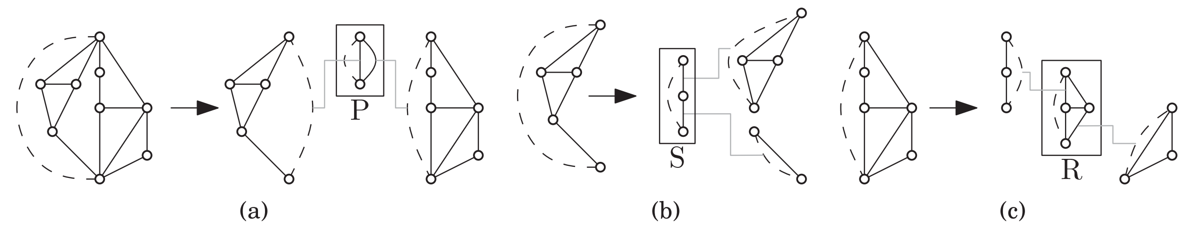


Fig. 4. Different cases in the construction of an SPQR-tree: parallel case (a), series case (b), and rigid case (c). The skeletons are shown in a box with the parent edge dashed. The dashed curves in the graphs represent the split pair with respect to which the graphs are decomposed; the procedure decomposes the graph without the dashed edge. The grey curves show the correspondence between the virtual edges of the skeleton and the corresponding split pairs in the children.

If μ has no children—that is, μ is a nonroot Q-node—then pert(μ) consists of the single edge represented by the node μ.

Note that the expansion graph of the parent edge of skel(μ) contains precisely those edges of G that do not belong to pert(μ). Note also that if ν is the parent of μ, then skel(ν) has a virtual edge whose expansion graph is pert(μ).

To give a precise definition of the SPQR-tree, we present an algorithm that recursively builds $\mathcal{T}$, starting from the root and then at each step adding a new child to a previously constructed node. Given a biconnected graph G and an edge $e = (u', v')$ of G, the algorithm proceeds as follows. At each recursive step, a split pair $\{u, v\}$ of G, a split component G^* of $\{u, v\}$, and a previously constructed node ν of $\mathcal{T}$ are given. A new node μ with pertinent graph G^* is added to $\mathcal{T}$ and attached as a child to ν. The vertices u and v are the *poles* of μ and are denoted by $u(\mu)$ and $v(\mu)$, respectively. The algorithm then possibly recurs on some split components of G^*.

At the beginning, the Q-node representing the edge $e = \{u', v'\}$ is designated as the root of $\mathcal{T}$. The algorithm then recurs with $G^* = G - \{e\}$, $\{u, v\} = \{u', v'\}$, and ν being the root. The recursive step distinguishes some cases, which are illustrated in Figure 4:

Base case: If G^* consists of exactly one edge between u and v, then μ is a Q-node whose skeleton is a cycle of length 2.

Parallel case: If G^* is composed of at least two maximal split components $G_1, \dots, G_k$ ($k \geq 2$) of G with respect to $\{u, v\}$, then μ is a P-node. The graph skel(μ) consists of k parallel virtual edges between u and v, denoted by $e_1, \dots, e_k$ and corresponding to $G_1, \dots, G_k$, respectively, plus an additional parent edge (u, v). The decomposition recurs on $G_1, \dots, G_k$, with $\{u, v\}$ as poles for every graph, and with μ as parent node.

Series case: If G^* is composed of exactly one maximal split component of G with respect to $\{u, v\}$ and if G^* has cutvertices $c_1, \dots, c_{k-1}$ ($k \geq 2$), appearing in this order on a path from u to v, then μ is an S-node. The graph skel(μ) is the cycle composed of a path $e_1, \dots, e_k$ of virtual edges, where e_1 connects u with c_1, e_i connects c_{i-1} with c_i ($i = 2, \dots, k-1$), and e_k connects c_{k-1} with v, plus a parent edge (u, v). The decomposition recurs on the split components corresponding to each of $e_1, e_2, \dots, e_{k-1}, e_k$ with μ as parent node and with $\{u, c_1\}, \{c_1, c_2\}, \dots, \{c_{k-2}, c_{k-1}\}, \{c_{k-1}, v\}$ as poles, respectively.

Rigid case: If none of the preceding cases applies, the purpose of the decomposition step is that of partitioning G^* into the minimum number of split components and recurring on each of them. We need some further definitions. Given a maximal split component G' of a split pair $\{s, t\}$ of G^*, a vertex $w \in G'$ *properly belongs* to G' if $w \neq s, t$. Given a split pair $\{s, t\}$ of G^*, a maximal split component G' of $\{s, t\}$ is *internal* if neither u nor v (the poles of G^*) properly belongs to G', *external* otherwise. A *maximal split pair* $\{s, t\}$ of G^* is a split pair of G^* that is not

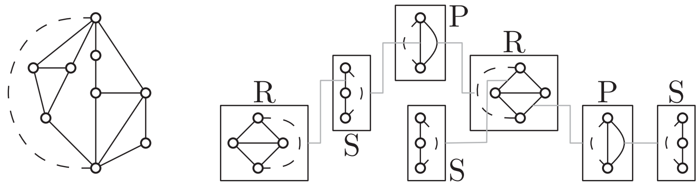


Fig. 5. The example graph of Figure 4 and its SPQR-tree with respect to the dashed reference edge. For clarity, Q-nodes are omitted. The parent edge of each skeleton is dashed, and each skeleton is connected by a grey curve to the virtual edge in the parent that represents it.

contained in an internal maximal split component of any other split pair $\{s', t'\}$ of G^*. Let $\{u_1, v_1\}, \ldots, \{u_k, v_k\}$ be the maximal split pairs of G^* ($k \geq 1$) that have at least one internal split component, and for $i = 1, \ldots, k$, let G_i be the union of all internal maximal split components of $\{u_i, v_i\}$. Observe that each vertex of G^* either properly belongs to exactly one G_i or belongs to some maximal split pair $\{u_i, v_i\}$. Node μ is an R-node. Graph skel(μ) is the graph obtained from G^* by replacing each subgraph G_i with the virtual edge e_i between u_i and v_i and by adding a parent edge (u, v). The decomposition recurs on each G_i with μ as parent node and with $\{u_i, v_i\}$ as poles. It can be shown that the skeleton of an R-node is a triconnected graph.

Figure 5 illustrates an example of the construction of the SPQR-tree of a biconnected graph G. The SPQR-tree $\mathcal{T}$ of an n-vertex biconnected graph G is well suited for the implementation of efficient algorithms, as $\mathcal{T}$ can be computed in linear time [Gutwenger and Mutzel 2000], it has $O(n)$ nodes, and the total size of all skeletons of all nodes of $\mathcal{T}$ is also $O(n)$ [Bertolazzi et al. 2000]. We say that an edge e of G *projects* to a virtual edge e' (or *belongs* to e') of skel(μ), for some node μ in $\mathcal{T}$, if e belongs to the expansion graph of e'.

The SPQR-tree $\mathcal{T}$ can be used to represent all planar embeddings of G. In the first part of the article, for our characterization, we will use the unrooted version of the SPQR-tree. In the second part of the article, for the linear-time implementation, we will then root the tree to allow for dynamic programming on the tree in a bottom-up fashion.

We emphasize the following properties, which are implicitly exploited throughout the article.

PROPERTY 1. *A planar embedding of the skeleton of every node of $\mathcal{T}$ determines a planar embedding of G and vice versa.*

PROPERTY 2. *Let C be a cycle of G and let μ be any node of $\mathcal{T}$. Then, either the edges of C belong to a single virtual edge of* skel(μ) *or they belong to a set of virtual edges that induce a cycle in* skel(μ).

Efficient computation of data structures. We now briefly discuss the time complexity of constructing the introduced data structures. We further show how to implement the basic queries used in our algorithms in constant time per operation.

First, observe that linear-time preprocessing can associate each edge of a planar graph with the unique connected component to which it belongs, with the unique block to which it belongs, and (given a planar embedding of the graph) with the (at most) two faces to which it is incident. Additionally, we can associate each vertex of a graph with the unique connected component to which it belongs.

The block-cutvertex tree of a connected planar graph [Tarjan 1972] and the SPQR-tree of a biconnected planar graph [Gutwenger and Mutzel 2000] can be constructed in linear time. The enriched block-cutvertex tree of a connected planar graph G can be constructed starting from the block-cutvertex tree of G by adding to the tree (i) each vertex v that is not a cutvertex of G, and (ii) an edge between v and the only block to which it belongs.

The block-face tree $\mathcal{BF}$ of a planar embedding $\mathcal{G}$ of a planar graph G can be constructed in linear time. Namely, for each edge e of G, let B_e be the unique block of G containing e and let f'_e and f''_e be the two faces of $\mathcal{G}$ adjacent to e (possibly $f'_e = f''_e$). Add edges (f'_e, B_e) and (f''_e, B_e) to $\mathcal{BF}$. When all edges of G have been considered, the resulting multigraph $\mathcal{BF}$ has a linear number of edges. Remove multiple edges as follows. Root $\mathcal{BF}$ at any node and orient $\mathcal{BF}$ so that all edges point toward the root. Remove all edges exiting from each node, except for one, thus obtaining the block-face tree $\mathcal{BF}$ of $\mathcal{G}$. The component-face tree $\mathcal{CF}$ of a planar embedding $\mathcal{G}$ of a planar graph G can be constructed analogously in linear time.

The vertex-face incidence graph $\mathcal{VF}$ of a planar embedding $\mathcal{G}$ of a planar graph G can be constructed in linear time by processing faces of $\mathcal{G}$ one by one, where for each face f we walk along the boundary of f and add to $\mathcal{VF}$ edges between f and the vertices on the boundary. To avoid adding multiple edges, we remember, for each vertex x, the last face f that has been connected to x in $\mathcal{VF}$. Note that $\mathcal{VF}$ is a planar graph.

Kowalik and Kurowski [2003] have shown that for a given planar graph F and a fixed integer k, it is possible to build in linear time a "short-path" data structure that allows checking in constant time whether two given vertices of F are connected by a path of length at most k and returning such a path if it exists. We will employ this data structure to search for paths of lengths 1 and 2 in our auxiliary graphs. Using this data structure, we can, for example, determine in constant time whether two vertices share a common face in $\mathcal{H}$ (by finding a path of length two in the vertex-face incidence graph $\mathcal{VF}$) or whether they share the same block (by finding a path of length 2 in the enriched block-cutvertex tree).

Efficient computation of local and nonlocal H-bridges. We now describe how these data structures can be used to solve the following problem. Given an instance $(G, H, \mathcal{H})$ of Pep and an H-bridge K of G, determine whether K is local or not and, in the latter case, compute the unique face f of $\mathcal{H}$ in which K has to be embedded in any solution of $(G, H, \mathcal{H})$. In the following lemma, we show how to solve this problem in time linear in the size of K.

Lemma 2.2. *Let $(G, H, \mathcal{H})$ be any instance of* Pep. *Assume that we are given the component-face tree $\mathcal{CF}$ of $\mathcal{H}$, the vertex-face incidence graph $\mathcal{VF}$ of $\mathcal{H}$, the block-face tree $\mathcal{BF}$ of $\mathcal{H}$, and, for each connected component C_i of H, the enriched block-cutvertex tree $\mathcal{B}_i^+$ of C_i. Suppose further that all of these graphs are endowed with short-path data structures to check for paths of length 1 and 2. Let K be an H-bridge of G. There is an algorithm that checks whether K is local to any block of H in time linear in the size of K. Furthermore, if K is nonlocal, the algorithm computes the only face of $\mathcal{H}$ incident to all attachment vertices of K, if such a face exists, in time linear in the size of K.*

Proof. Consider the attachment vertices $a_1, a_2, \ldots, a_h$ of K. If $h = 1$, then K is local. Otherwise, $h \geq 2$. To decide whether K is local for some block of H, we perform the following check. Consider the attachment vertices a_1 and a_2. If a_1 and a_2 belong to distinct connected components, then K is not local to any block. Otherwise, they belong to the same connected component C_i. Check whether a_1 and a_2 have distance 2 in $\mathcal{B}_i^+$—that is, whether they belong to the same block B. This is done in constant time by querying the short-path structure. If the check fails, then K is not local to any block. Otherwise, B contains both a_1 and a_2. In the latter case, check whether B is also adjacent in $\mathcal{B}_i^+$ to all other attachment vertices $a_3, \ldots, a_h$ of K. Again, each such a check is performed in constant time. If the test succeeds, then K is local to block B.

Otherwise, there exists a vertex a_j, with $3 \leq j \leq h$, that is not incident to B, and K is not local to any block.

If K is nonlocal, we compute the unique face f of $\mathcal{H}$ to which all attachment vertices of K are incident. First, we choose two attachment vertices a_p and a_q, with $1 \leq p, q \leq h$, that do not belong to the same block. If a_1 and a_2 do not belong to the same block, then we take $a_p = a_1$ and $a_q = a_2$. If the check failed on an attachment vertex a_j in $a_3, \dots, a_h$, then either a_1 and a_j or a_2 and a_j do not belong to the same block. In the former case, set $a_p = a_1$ and $a_q = a_j$; in the latter one, set $a_p = a_2$ and $a_q = a_j$. By querying the short-path data structure, we determine in constant time whether a_p and a_q are connected by a path of length 2 in $\mathcal{V}F$ and find the middle vertex of such a path. This middle vertex corresponds to the unique common face f of a_p and a_q. We then check whether all attachments of K are adjacent to f in $\mathcal{V}F$. If the test fails, then no face of $\mathcal{H}$ contains all attachments of K. Otherwise, f is the only face of $\mathcal{H}$ whose boundary contains all attachments of K. □

3. COMBINATORIAL CHARACTERIZATION

We first present a combinatorial characterization of planar PEGS. This not only forms a basis of our algorithm but also is interesting in its own right, as it shows that a PEG has an embedding extension if and only if it satisfies simple conditions that are obviously necessary for an embedding extension to exist.

Our characterization is based on a decomposition of the graph G of a PEG $(G, H, \mathcal{H})$ into its connected, biconnected and triconnected components. For triconnected PEGS, the problem is particularly easy. For a triconnected PEG $(G, H, \mathcal{H})$, the graph G has only two distinct planar embeddings: $\mathcal{G}_1$ and $\mathcal{G}_2$. The PEG is thus planar if and only if either $\mathcal{G}_1$ or $\mathcal{G}_2$ extends $\mathcal{H}$. Clearly, for a disconnected PEG to admit an embedding extension, it is a necessary condition that each of its connected components admits an embedding extension. Similarly, it is a necessary condition for a connected PEG that each biconnected component admits an embedding extension. We start with the most specific case—the case where G is biconnected—and then extend the characterization to the cases where G is connected or even disconnected.

3.1. Planarity of Biconnected PEGS

In this section, we focus on biconnected PEGS $(G, H, \mathcal{H})$. This assumption allows us to use the SPQR-tree $\mathcal{T}$ of G as the main tool of our characterization, which is based on the two necessary and sufficient conditions of Lemma 2.1. We show that they can be individually translated to constraints on the embeddings of the skeletons of $\mathcal{T}$.

Definition 3.1. A planar embedding of the skeleton of a node μ of the SPQR-tree of G is *edge-compatible with* $\mathcal{H}$ if, for every vertex x of skel(μ), and for every three edges of $E_H(x)$ belonging to different virtual edges of skel(μ), their clockwise order determined by the embedding of skel(μ) is a suborder of $\sigma_{\mathcal{H}}(x)$.

LEMMA 3.2. *Let $(G, H, \mathcal{H})$ be a biconnected* PEG. *Let $\mathcal{T}$ be the SPQR-tree of G. An embedding $\mathcal{G}$ of G satisfies condition 1 of Lemma 2.1 if and only if for each node μ of $\mathcal{T}$, the corresponding embedding of* skel(μ) *is edge-compatible with $\mathcal{H}$.*

PROOF. Obviously, if G has an embedding satisfying condition 1 of Lemma 2.1, then the corresponding embedding of skel(μ) is edge-compatible with $\mathcal{H}$ for each node μ of $\mathcal{T}$.

To prove the converse, assume that the skeleton of every node of $\mathcal{T}$ has an embedding that is edge-compatible with $\mathcal{H}$, and let $\mathcal{G}$ be the embedding of G determined by all such skeleton embeddings. We claim that $\mathcal{G}$ satisfies condition 1 of Lemma 2.1. To prove the claim, it suffices to show that any three edges e, f, and g of $\mathcal{H}$ that share a common

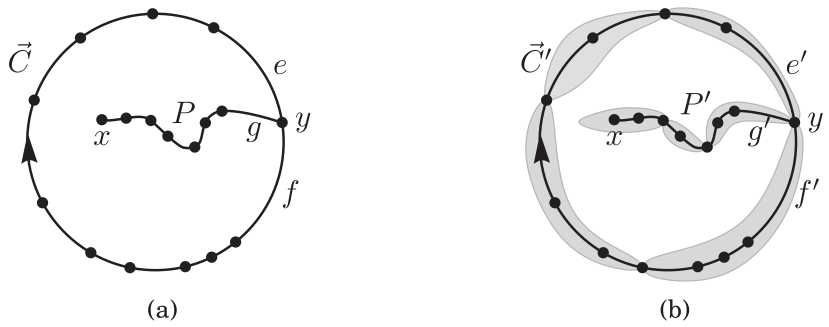


Fig. 6. Illustration for the proof of Lemma 3.4. (a) Path P, vertices x and y, and edges e, f, and g in $\mathcal{G}$. (b) Path or cycle P' and edges e', f', and g' of skel(μ). Grey regions represent virtual edges of the skeleton of a node of $\mathcal{T}$.

vertex x appear in the same clockwise order around x in $\mathcal{H}$ and in $\mathcal{G}$. Assume that the triple (e, f, g) is embedded in clockwise order around x in $\mathcal{H}$. Let μ be the node of $\mathcal{T}$ with the property that the Q-nodes representing e, f, and g appear in distinct components of $\mathcal{T} - \mu$. Note that such a node μ exists and is unique. The three edges e, f, and g project into three distinct virtual edges e', f', and g' of skel(μ). Since the embedding of skel(μ) is assumed to be edge-compatible with $\mathcal{H}$, the triple (e', f', g') is embedded in clockwise order in skel(μ), and hence the triple (e, f, g) is embedded in clockwise order in $\mathcal{G}$. □

Lemma 3.2 settles the translation of condition 1 of Lemma 2.1 to conditions on the embeddings of the skeletons of the SPQR-tree of G. Next, we deal with condition 2. Consider a simple cycle $\vec{C}$ of G with an arbitrary orientation and a node μ of the SPQR-tree of G. By Property 2, either all edges of $\vec{C}$ belong to the expansion graph of a single virtual edge of skel(μ) or the virtual edges whose expansion graphs contain the edges of $\vec{C}$ form a simple cycle in skel(μ). Such a cycle in skel(μ) inherits the orientation of $\vec{C}$ in a natural way.

Definition 3.3. A planar embedding of the skeleton of a node μ of the SPQR-tree of G is *cycle-compatible with* $\mathcal{H}$ if for every facial cycle $\vec{C}$ of $\mathcal{H}$ whose edges project to a simple cycle $\vec{C}'$ in skel(μ), all vertices of skel(μ) that belong to $V_{\mathcal{H}}^{\text{left}}(\vec{C})$ and all virtual edges that contain vertices of $V_{\mathcal{H}}^{\text{left}}(\vec{C})$ (except for the virtual edges of $\vec{C}'$ itself) are embedded to the left of $\vec{C}'$, and analogously for $V_{\mathcal{H}}^{\text{right}}(\vec{C})$.

LEMMA 3.4. *Let* $(G, H, \mathcal{H})$ *be a biconnected* PEG. *Let* $\mathcal{T}$ *be the SPQR-tree of* G. *An embedding* $\mathcal{G}$ *of* G *satisfies condition 2 of Lemma 2.1 if and only if for each node* μ *of* $\mathcal{T}$, *the corresponding embedding of* skel(μ) *is cycle-compatible with* $\mathcal{H}$.

PROOF. Obviously, if $\mathcal{G}$ is an embedding of G that satisfies condition 2 of Lemma 2.1, then the corresponding embedding of skel(μ) is cycle-compatible with $\mathcal{H}$ for each node μ of $\mathcal{T}$.

To prove the converse, assume that skel(μ) has an embedding that is cycle-compatible with $\mathcal{H}$ for each node μ of $\mathcal{T}$, and let $\mathcal{G}$ be the resulting embedding of G.

Our goal is to show that for every facial cycle $\vec{C}$ of $\mathcal{H}$ and for every vertex x of $H - V(\vec{C})$, the left/right position of x with respect to $\vec{C}$ is the same in $\mathcal{H}$ as in $\mathcal{G}$.

Refer to Figure 6(a). Assume that x is to the right of $\vec{C}$ in $\mathcal{G}$ (the other case being analogous). Let P be a shortest path in G that connects x to a vertex of $\vec{C}$. Such a path exists since G is connected. Let y be the vertex of $\vec{C} \cap P$, and let e and f be the two edges of $\vec{C}$ adjacent to y, where e directly precedes f in the orientation of $\vec{C}$. By the minimality of P, all vertices of $P - y$ avoid $\vec{C}$, and hence all vertices of $P - y$ are to the right of $\vec{C}$ in $\mathcal{G}$. Let g be the edge of P adjacent to y. In $\mathcal{G}$, the triple (e, f, g) appears in clockwise order around y.

Refer to Figure 6(b). Let μ be the (unique) internal node of $\mathcal{T}$ in which e, f, and g project to distinct edges e', f', and g' of skel(μ). Let $\vec{C}'$ be the projection of $\vec{C}$ into skel(μ) (i.e., $\vec{C}'$ is the subgraph of skel(μ) formed by edges that contain the projection of at least one edge of $\vec{C}$), and let P' be the projection of P. It is easy to see that $\vec{C}'$ is a cycle of length at least 2, whereas P' is either a path or a cycle. Note that the latter case only happens when P' has at least two edges and the vertex x properly belongs to a virtual edge d' of skel(μ) incident with y. Assume that the edges of $\vec{C}'$ are oriented consistently with the orientation of $\vec{C}$ and that the edges of P' form an ordered sequence, where the edge containing x is the first and g' is the last.

Both the endpoints of an edge of $\vec{C}'$ are vertices of $\vec{C}$. Analogously, both the endpoints of an edge of P' are vertices of P, with the possible exception of the first vertex of P'. It follows that no vertex of P' belongs to $\vec{C}'$, except possibly for the first one and the last one. Thus, no edge of P' belongs to $\vec{C}'$, and by the assumption that the embedding of skel(μ) is planar and that $\mathcal{G}$ is the embedding resulting from the skeleton embedding choices, all edges of P' are embedded to the right of the directed cycle $\vec{C}'$ in skel(μ). In particular, the edge of skel(μ) containing x is to the right of $\vec{C}'$. Since the embedding of skel(μ) is assumed to be cycle-compatible with $\mathcal{H}$, x is to the right of $\vec{C}$ in $\mathcal{H}$.

This shows that $\mathcal{G}$ satisfies condition 2 of Lemma 2.1, as claimed. □

Definition 3.5. A planar embedding of the skeleton of a node μ of the SPQR-tree of G is *compatible with* $\mathcal{H}$ if it is both edge- and cycle-compatible with $\mathcal{H}$.

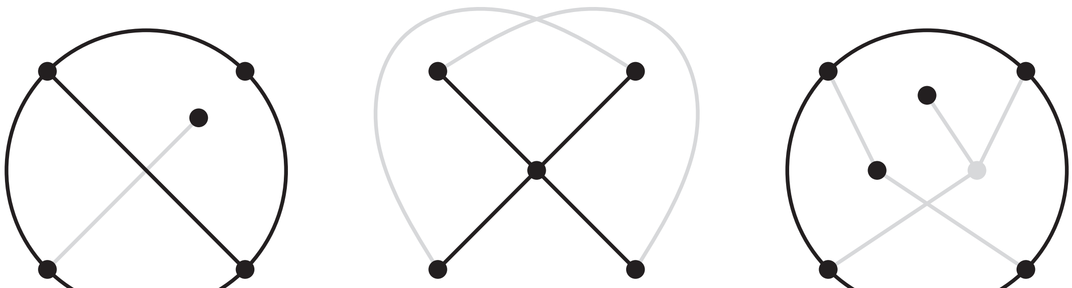
As a consequence of Lemmas 3.2 and 3.4, we obtain the following characterization of planar biconnected PEGs.

THEOREM 3.6. *Let* $(G, H, \mathcal{H})$ *be a biconnected* PEG. *Then, G has an embedding that extends $\mathcal{H}$ if and only if the skeleton of each node of its SPQR-tree has an embedding compatible with $\mathcal{H}$.*

If G is biconnected, we can use Theorem 3.6 for devising a polynomial-time algorithm for PEP. Namely, we can test, for each node μ of the SPQR-tree $\mathcal{T}$ of G, whether skel(μ) has an embedding that is compatible with $\mathcal{H}$. For Q-, S-, and R-nodes, this test is easily done in polynomial time.

If μ is a P-node, the test is more complex. Let x and y be the two poles of skel(μ). We say that a virtual edge e of skel(μ) is *constrained* if the expansion graph of e contains at least one edge of H incident to x and at least one edge of H incident to y. To obtain an embedding of μ edge-compatible with $\mathcal{H}$, the constrained edges must be embedded in a cyclic order that is consistent with $\sigma_{\mathcal{H}}(x)$ and $\sigma_{\mathcal{H}}(y)$. Such a cyclic order, if it exists, is unique and can be determined in polynomial time. Note that if $\mathcal{H}$ has a facial cycle $\vec{C}$ that projects to a proper cycle $\vec{C}'$ in μ, then $\vec{C}'$ has exactly two edges and these two edges are both constrained. Thus, the embedding of any such cycle $\vec{C}'$ in μ is fixed as soon as we fix the cyclic order of the constrained edges. Once the cyclic order of the constrained edges of μ is determined, we process the remaining edges one by one and insert them among the edges that are already embedded in such a way that no edge- or cycle compatibility constraints are violated. It is not difficult to verify that this procedure constructs an embedding of μ compatible with $\mathcal{H}$, if such an embedding exists.

Thus, PEP can be solved in polynomial time for biconnected PEGs.



(a) (b) (c)

Fig. 7. Three examples of PEP instances $(G, H, \mathcal{H})$ that have no embedding extension, even though each block of G admits an embedding extending the corresponding sub-embedding of $\mathcal{H}$. The black edges and vertices represent $\mathcal{H}$, and the grey edges and vertices belong to G but not to H. Note that instance (a) fails to satisfy condition 3 of Lemma 3.8 (shown later), instance (b) fails to satisfy condition 2 of Lemma 3.8, and instance (c) has a nontrivial nonlocal H-bridge. The modification of instance (c) into an equivalent instance without nontrivial nonlocal H-bridges creates a block of G that does not have an embedding extension.

3.2. Planarity of Connected and Disconnected PEGs

A graph is planar if and only if each of its blocks is planar. Thus, planarity testing of general graphs can be reduced to planarity testing of biconnected graphs. For planarity testing of partially embedded graphs, the same simple reduction does not work (Figure 7). However, we will show that solving partially embedded planarity for a general instance $(G, H, \mathcal{H})$ can be reduced to solving the subinstances induced by the blocks of G and to checking additional conditions guaranteeing that the partial solutions can be combined into a full solution for $(G, H, \mathcal{H})$.

Let us consider a connected PEG $(G, H, \mathcal{H})$—that is, an instance of PEP in which G is connected. When dealing with such an instance, it is useful to assume that G has no nontrivial nonlocal H-bridge. We will now show that any instance of PEP can be transformed into an equivalent instance that satisfies this additional assumption.

Let K be a nontrivial nonlocal H-bridge of G. Since K is nonlocal, it must have at least two attachments that do not belong to any single block of H. Let f_K be the face of $\mathcal{H}$ whose boundary contains all attachments of the H-bridge K, if any such a face exists. Otherwise, let f_K be an arbitrary face of $\mathcal{H}$.

Let $\mathcal{K}$ be the set of nontrivial nonlocal H-bridges of G. It is clear that in any embedding of G extending $\mathcal{H}$, all vertices of $K - V(H)$ are embedded inside f_K for every $K \in \mathcal{K}$. This motivates the following definition.

Definition 3.7. Let H' be the graph whose edge set is equal to the edge set of H and whose vertex set is defined by $V(H') = V(H) \cup \bigcup_{K \in \mathcal{K}} V(K)$. Let $\mathcal{H}'$ be the embedding of H' that is obtained from $\mathcal{H}$ by inserting, for every H-bridge $K \in \mathcal{K}$, all vertices of $K - V(H)$ into the interior of face f_K.

Observe that the graph G has no nontrivial nonlocal H'-bridges. In addition, observe that any embedding of G that extends $\mathcal{H}$ also extends $\mathcal{H}'$, and vice versa. Thus, the instance $(G, H, \mathcal{H})$ of PEP is equivalent to the instance $(G, H', \mathcal{H}')$, which contains no nontrivial nonlocal bridges.

Before we state the next lemma, we need more terminology. Let $\mathcal{H}$ be an embedding of a graph H, and let H_1 and H_2 be edge-disjoint subgraphs of H. We say that H_1 and H_2 *alternate* around a vertex x of $\mathcal{H}$ if there are two pairs of edges $e, e' \in E(H_1)$ and $f, f' \in E(H_2)$ that are incident to x and that appear in the cyclic order (e, f, e', f') in the rotation of x restricted to these four edges. Let x and y be two vertices of H and let $\vec{C}$ be a directed cycle in $\mathcal{H}$. We say that $\vec{C}$ *separates* x and y if $x \in V^{\text{left}}_{\mathcal{H}}(\vec{C})$ and $y \in V^{\text{right}}_{\mathcal{H}}(\vec{C})$, or vice versa.

LEMMA 3.8. *Let $(G, H, \mathcal{H})$ be an instance of* PEP *where G is connected and every nontrivial H-bridge of G is local. Let $G_1, \dots, G_t$ be the blocks of G, let H_i be the subgraph of H induced by the vertices of G_i, and let $\mathcal{H}_i$ be $\mathcal{H}$ restricted to H_i. Then, G has an embedding extending $\mathcal{H}$ if and only if*

(1) *G_i has an embedding extending $\mathcal{H}_i$, for every $1 \leq i \leq t$,*
(2) *no two distinct graphs H_i and H_j alternate around any vertex of $\mathcal{H}$, and*
(3) *for every facial cycle $\vec{C}$ of $\mathcal{H}$ and for any two vertices x and y of $\mathcal{H}$ separated by $\vec{C}$, any path in G connecting x and y contains a vertex of $\vec{C}$.*

PROOF. Clearly, the three conditions of the lemma are necessary. To show that they are also sufficient, assume that the three conditions are satisfied and proceed by induction on the number t of blocks of G.

If $t = 1$, then G is biconnected and there is nothing to prove. Assume that $t \geq 2$. If there is at least one block G_i that does not contain any vertex of H, we consider the subgraph G' of G consisting of those blocks that contain at least one vertex of H. Since every nontrivial H-bridge of G is local, the graph G' is connected, and hence it satisfies the three conditions of the lemma. By induction, the embedding $\mathcal{H}$ can be extended into an embedding $\mathcal{G}'$ of G'. Since every block G_i of G is planar (by condition 1 of the lemma), it is easy to extend the embedding $\mathcal{G}'$ into an embedding $\mathcal{G}$ of G.

Assume now that every block of G contains at least one vertex of H. This implies that every cutvertex of G belongs to H, because otherwise the cutvertex would belong to a nontrivial nonlocal H-bridge, which is impossible by assumption. Let x be any cutvertex of G. Let $G'_1, G'_2, \dots, G'_k$ be the connected components of $G - x$, where we select G'_1 by the following rules: if there is a component of $G - x$ that has no vertex connected to x by an edge of H, then let G'_1 be such a component; if each component of $G - x$ is connected to x by an edge of H, then choose G'_1 in such a way that the edges of H incident to x and belonging to G'_1 form an interval in $\sigma_{\mathcal{H}}(x)$. Such a choice of G'_1 is always possible, due to condition 2 of the lemma.

Let G' be the subgraph of G induced by $V(G'_1) \cup \{x\}$ and let G'' be the subgraph of G induced by $V(G'_2) \cup \dots \cup V(G'_k) \cup \{x\}$. Let H' and H'' be the subgraphs of H induced by the vertices of G' and G'', respectively, and let $\mathcal{H}'$ and $\mathcal{H}''$ be $\mathcal{H}$ restricted to H' and H'', respectively. Both G' and G'' have fewer blocks than G. In addition, both instances, $(G', H', \mathcal{H}')$ and $(G'', H'', \mathcal{H}'')$, satisfy the conditions of the lemma. Thus, by induction, there is an embedding $\mathcal{G}'$ of G' that extends $\mathcal{H}'$ and an embedding $\mathcal{G}''$ of G'' that extends $\mathcal{H}''$. Our goal is to combine $\mathcal{G}'$ and $\mathcal{G}''$ into a single embedding of G that extends $\mathcal{H}$. To see that this is possible, we prove two auxiliary claims.

Claim 1. $\mathcal{H}'$ has a face f' whose boundary contains x and, for any facial cycle $\vec{C}$ of f', all vertices of H'' except for x are in $V^{\text{left}}_{\mathcal{H}}(\vec{C})$—that is, they are "inside" f'.

To see that the claim holds, assume first that H' has no edge incident to x (Figure 8(a)). Let f' be the unique face of $\mathcal{H}'$ incident to x. We show that all vertices of H'' are inside f' in $\mathcal{H}$. Let y be any vertex of H''. Since G'' is connected, there is a path P in G'' from y to x. Assume for contradiction that $\mathcal{H}'$ has a facial cycle $\vec{C}$ such that $\vec{C}$ separates y from x in $\mathcal{H}$. This cycle belongs to $H' - x$, and hence $\vec{C}$ and P are disjoint, contradicting condition 3 of the lemma.

Next, assume that H' has an edge incident to x (see Figure 8(b)). By the construction of G_1, each connected component of $G - x$ has at least one vertex connected to x by an edge of H. Moreover, the edges of $\mathcal{H}'$ incident to x form an interval in $\sigma_{\mathcal{H}}(x)$. This shows that $\mathcal{H}'$ has a face f' containing x on its boundary, such that every vertex of H''

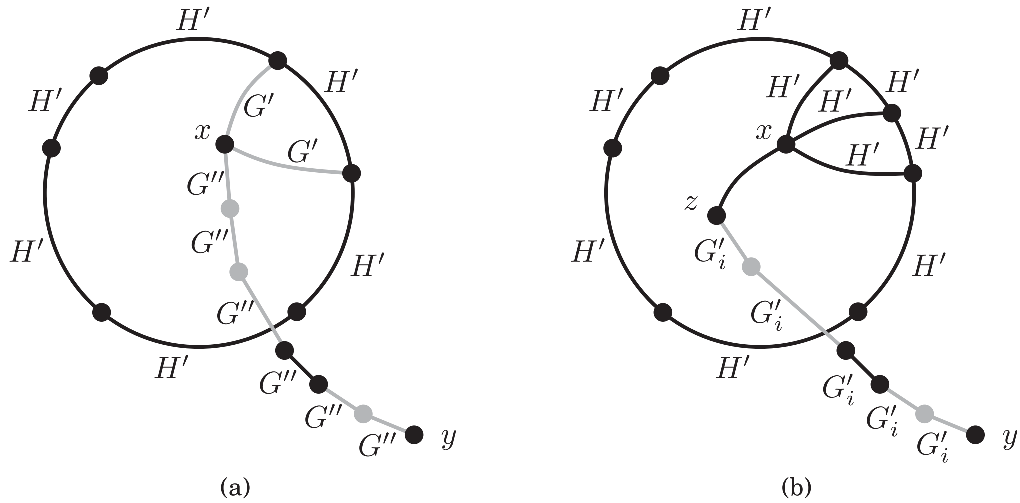


Fig. 8. Illustration for the proof of Lemma 3.8. (a) H' has no edge incident to x. (b) H' has an edge incident to x.

adjacent to x is inside f' in $\mathcal{H}$. We now show that all vertices of H'' except for x are inside f'. Let y be a vertex of H'' different from x. Let G'_i be the component of $G - x$ containing y. We know that G'_i has a vertex z adjacent to x by an edge of H and that z is inside f' in $\mathcal{H}$. Let P be a path in G'_i connecting y and z. If y is not inside f', then y is separated from z in $\mathcal{H}$ by a facial cycle of $\mathcal{H}'$, contradicting condition 3 of the lemma.

Claim 2. All vertices of H', except for x, appear in $\mathcal{H}$ inside the same face f'' of $\mathcal{H}''$; furthermore, x is on the boundary of f''.

To prove the claim, note that any two vertices from $H' - x$ are inside the same face f'' of $\mathcal{H}''$ in $\mathcal{H}$ by condition 3 of the lemma because they are connected by a path in G'_1. Vertex x is on the boundary of f'', as otherwise it would be separated in $\mathcal{H}$ from the remaining vertices of H' by a facial cycle of f'', again contradicting condition 3 of the lemma.

In view of the previous two claims, the embedding $\mathcal{G}'$ of G' and the embedding $\mathcal{G}''$ of G'' can be combined into a single embedding $\mathcal{G}$ of G that extends $\mathcal{H}$. To see this, note that when $\mathcal{H}'$ is extended into $\mathcal{G}'$, the face f' from Claim 1 can be subdivided into several faces of $\mathcal{G}'$, at least one of which, say g', contains x on its boundary. Analogously, the face f'' from Claim 2 can be subdivided into several faces of $\mathcal{G}''$, at least one of which, say g'', contains x on its boundary. We then obtain the embedding $\mathcal{G}$ by merging the faces g' and g'' into a single face. □

Observe that computing the nonlocal H-bridges $\mathcal{K}$ and, for each nonlocal H-bridge $K \in \mathcal{K}$, the face f_K can be done in polynomial time. Afterward, the second and third conditions of Lemma 3.8 can easily be checked in polynomial time.

Next, we focus on disconnected PEGS—that is, the instances $(G, H, \mathcal{H})$ of PEP in which G is not connected. The possibility of solving the subinstances of $(G, H, \mathcal{H})$ induced by the connected components of G does not guarantee that the instance $(G, H, \mathcal{H})$ of PEP has a solution. However, we show that solving PEP for an instance $(G, H, \mathcal{H})$ can be reduced to solving the subinstances induced by the connected components of G and to checking additional conditions that guarantee that the partial solutions can be combined into a full solution for $(G, H, \mathcal{H})$.

LEMMA 3.9. *Let* $(G, H, \mathcal{H})$ *be an instance of* PEP. *Let* $G_1, \ldots, G_t$ *be the connected components of* G. *Let* H_i *be the subgraph of* H *induced by the vertices of* G_i, *and let* $\mathcal{H}_i$ *be* $\mathcal{H}$ *restricted to* H_i. *Then* G *has an embedding extending* $\mathcal{H}$ *if and only if*

(1) *G_i has an embedding extending $\mathcal{H}_i$, for every $1 \le i \le t$, and*
(2) *for every $1 \le i, j \le t$ with $j \ne i$, and for each facial cycle $\vec{C}$ of H_i, no two vertices of H_j are separated by $\vec{C}$.*

Proof. Clearly, the two conditions of the lemma are necessary. To show that they are also sufficient, assume that the two conditions are satisfied and proceed by induction on the number t of connected components of G.

If $t = 1$, then G is connected and there is nothing to prove. Assume now that G has $t \ge 2$ connected components $G_1, \dots, G_t$. Let H_i and $\mathcal{H}_i$ be defined as in the statement of the lemma. Note that H_i may consist of several connected components. Let $\mathcal{CF}$ be the component-face tree of $\mathcal{H}$, rooted at a node that represents an arbitrary face of $\mathcal{H}$. We say that a face f_i of $\mathcal{H}$ is the *outer face* of $\mathcal{H}_i$ if at least one child of f_i in $\mathcal{CF}$ is a component of H_i but the parent of f_i is not a component of H_i. Observe that due to the second condition of the lemma, each $\mathcal{H}_i$ has exactly one outer face f_i. We thus have a sequence of (not necessarily distinct) outer faces $f_1, \dots, f_t$ of $\mathcal{H}_1, \dots, \mathcal{H}_t$.

Let us now assume, without loss of generality, that in the subtree of $\mathcal{CF}$ rooted at f_1, there is no outer face $f_i \ne f_1$. This implies that f_1 is the only face of $\mathcal{H}$ that is incident both to H_1 and to $H - H_1$. By induction, the embedding $\mathcal{H} - \mathcal{H}_1$ can be extended to an embedding $\mathcal{G}_{\ge 2}$ of the graph $G - G_1$. By the first condition of the lemma, $\mathcal{H}_1$ can be extended into an embedding $\mathcal{G}_1$ of G_1. The two embeddings $\mathcal{H} - \mathcal{H}_1$ and $\mathcal{H}_1$ share a single face f_1.

When extending the embedding $\mathcal{H}_1$ into $\mathcal{G}_1$, the face f_1 of $\mathcal{H}_1$ can be subdivided into several faces of $\mathcal{G}_1$. Let f' be any face of $\mathcal{G}_1$ obtained by subdividing f_1. Analogously, in the embedding $\mathcal{G}_{\ge 2}$ the face f_1 can be subdivided into several faces, among which we choose an arbitrary face f''.

We then glue the two embeddings $\mathcal{G}_1$ and $\mathcal{G}_{\ge 2}$ by identifying the face f' of $\mathcal{G}_1$ and the face f'' of $\mathcal{G}_{\ge 2}$ into a single face whose boundary is the union of the boundaries of f' and f''. This yields an embedding of G that extends $\mathcal{H}$. □

Note that the second condition of Lemma 3.9 can easily be tested in polynomial time. Thus, we can use the characterization to directly prove that Partially Embedded Planarity is solvable in polynomial time. In the rest of the article, we describe a more sophisticated algorithm that solves Pep in linear time.

4. LINEAR-TIME ALGORITHM

In this section, we devise a linear-time algorithm for solving Pep. The algorithm basically follows the outline of the characterization. The first milestone is a linear-time algorithm for testing the planarity of biconnected Pegs. Afterward, we show that the additional conditions for the planarity of connected and disconnected Pegs can be checked in linear time.

Essentially, to solve the biconnected case, it is sufficient to give a linear-time implementation of the algorithm sketched at the end of Section 3.1. In fact, most of the steps sketched there are fairly easy to implement in linear time. The problem of finding a compatible embedding of a P-node, however, is tricky. Indeed, a P-node μ may contain a linear number of facial cycles of $\mathcal{H}$ that project to cycles in skel(μ). Further, a linear number of virtual edges of skel(μ) may have no H-edge adjacent to the poles of μ; hence, the positions at which they have to be inserted in the cyclic orderings around the poles of μ depend only on the cycle containment constraints. To process the skeleton of a P-node μ in time proportional to its size, we would need to find, for each virtual edge e of skel(μ), a position such that e is contained in all and only the cycles in which it needs to be contained.

Therefore, the main problem for reaching linear running time stems from the cycle compatibility constraints (condition 2 of Lemma 2.1). The constraints stemming from rotation system (condition 1 of Lemma 2.1) consist of orderings of subsets of the edges incident to each vertex. Thus, the total size of the latter constraints is linear, and additionally, the constraints are very "local." In light of these two properties, it is not surprising that the rotation system constraints are relatively simple to handle in total linear time. The same does not hold, however, for the cycle containment constraints. As specified in condition 2 of Lemma 2.1, these constraints determine, for each directed facial cycle $\vec{C}$ of H and each vertex v of $H - C$, whether v is to the left or to the right of $\vec{C}$. Note that the graph H may contain a linear number of facial cycles, and thus this amounts to quadratically many cycle–vertex constraints. Further, these constraints do not exhibit any locality on G. Evidently, a lot of the information encoded in the cycle containment constraints is redundant, as the set of cycles involved in these constraints is the set of facial cycles of a planar graph.

We use two different approaches to handle the cycle containment constraints in linear time. One is to ignore them; we prove that this yields a correct solution if H is connected, as in this case the cycle containment constraints are implied by the rotation system constraints. The second consists of considering restricted instances, where the constraints can easily be expressed in linear time and space. Suppose that we have a Peg $(G, H, \mathcal{H})$ with a face f of $\mathcal{H}$ such that all vertices of H are part of at least one facial cycle of f. This implies that each facial cycle of H has f on its left side and the right side does not contain any vertex of H. In this case, the cycle containment constraints of each facial cycle $\vec{C}$ of f can be expressed as $V_{\mathcal{H}}^{\text{right}}(\vec{C}) = \emptyset$, thus yielding a set of constraints whose size is linear. Moreover, in the SPQR-tree, it is sufficient to keep track of which virtual edges contain vertices of H and which do not. This information is much easier to aggregate than information about individual vertices and all of their cycle containment constraints.

First, we tackle the case in which G is biconnected. The algorithm solving this case, presented in Section 4.3, uses the algorithms presented in Sections 4.1 and 4.2 as subroutines to solve more restricted subcases. Then, we deal with the case in which G is simply connected and with the general case, where G may be disconnected, in Section 4.4. The algorithm we present exploits several auxiliary data structures, namely block-cutvertex trees, SPQR-trees, enriched block-cutvertex trees, block-face trees, component-face trees, and vertex-face incidence graphs. Note that all of these data structures can easily be computed in linear time (see Section 2.3).

4.1. *G* Biconnected, *H* Connected

In this section, we show how to solve Pep in linear time for biconnected Pegs $(G, H, \mathcal{H})$ with H connected. We first show that in this case the rotation system alone is sufficient for finding an embedding extension.

Lemma 4.1. *Let $(G, H, \mathcal{H})$ be a* Peg *such that H is connected. Let $\mathcal{G}$ be any planar embedding of G satisfying condition 1 of Lemma 2.1. Then, $\mathcal{G}$ satisfies condition 2 of Lemma 2.1.*

Proof. Suppose, for a contradiction, that a planar embedding $\mathcal{G}$ of G exists such that $\mathcal{G}$ satisfies condition 1 and does not satisfy condition 2 of Lemma 2.1. Then, there exists a facial cycle $\vec{C}$ of $\mathcal{H}$ such that either there exists a vertex $x \in V_{\mathcal{H}}^{\text{left}}(\vec{C})$ with $x \in V_{\mathcal{G}}^{\text{right}}(\vec{C})$ or there exists a vertex $x \in V_{\mathcal{H}}^{\text{right}}(\vec{C})$ with $x \in V_{\mathcal{G}}^{\text{left}}(\vec{C})$. Suppose that we are in the former case, as the latter case can be treated analogously. Since $\mathcal{H}$ is a planar embedding and H is connected, there exists a path $P = (x_1, x_2, \ldots, x_k) \in H$ such that x_1

is a vertex of $\vec{C}$, $x_i \in V^{\text{left}}_{\mathcal{H}}(\vec{C})$, for each $i = 2, \dots, k$, and $x_k = x$. Denote by x_1^- and by x_1^+ the vertex preceding and following x_1 in the oriented cycle $\vec{C}$, respectively. Consider the placement of x_2 with respect to $\vec{C}$ in $\mathcal{G}$. As $x_2 \notin \vec{C}$, either $x_2 \in V^{\text{left}}_{\mathcal{G}}(\vec{C})$ or $x_2 \in V^{\text{right}}_{\mathcal{G}}(\vec{C})$. In the first case, the path $(x_2, \dots, x_k)$ crosses $\vec{C}$, since $x_2 \in V^{\text{left}}_{\mathcal{G}}(\vec{C})$, $x_k \in V^{\text{right}}_{\mathcal{G}}(\vec{C})$, and no vertex v_i belongs to $\vec{C}$, for $i = 2, \dots, k$, thus contradicting the planarity of the embedding $\mathcal{G}$. In the second case, the clockwise order of the edges incident to x_1 in $\mathcal{H}$ is (x_1, x_1^-), (x_1, x_2), and (x_1, x_1^+), whereas the clockwise order of the edges incident to x_1 in $\mathcal{G}$ is (x_1, x_1^-), (x_1, x_1^+), and (x_1, x_2), thus contradicting the assumption that $\mathcal{G}$ satisfies condition 1 of Lemma 2.1. □

By Lemma 4.1, testing whether a planar embedding $\mathcal{G}$ exists satisfying conditions 1 and 2 of Lemma 2.1 is equivalent to testing whether a planar embedding $\mathcal{G}$ exists satisfying condition 1 of Lemma 2.1. Due to Lemma 3.2, testing whether a planar embedding $\mathcal{G}$ exists satisfying condition 1 is equivalent to testing whether the skeleton of each node of the SPQR-tree of G has a planar embedding that is edge-compatible with $\mathcal{H}$. We now describe an algorithm—Algorithm BC (for G Biconnected and H Connected)—that achieves this in linear time.

Algorithm BC. Construct the SPQR-tree $\mathcal{T}$ of G and root it at an arbitrary Q-node. This choice determines the pertinent graph of each node μ of $\mathcal{T}$ and also determines the parent virtual edge in the skeleton of each nonroot node μ of $\mathcal{T}$. A bottom-up visit of $\mathcal{T}$ is performed, such that after a node μ of $\mathcal{T}$ has been visited, an embedding of skel(μ) that is edge-compatible with $\mathcal{H}$ is selected, if it exists. To find an edge-compatible embedding of the skeleton skel(μ) of a node μ of $\mathcal{T}$, we need to know whether the expansion graph of each virtual edge uv of skel(μ) contains H-edges incident to u and v. Due to the bottom-up traversal, the pertinent graphs of all children have already been processed by the algorithm, and we can thus aggregate this information for all virtual edges, except for the parent edge.

To keep track of the edges of H that belong to pert(μ) and that are incident to the pole $u(\mu)$, define the *first edge* $f_{u(\mu)}$ and the *last edge* $l_{u(\mu)}$ as the edges of H incident to $u(\mu)$ in pert(μ) such that all other edges of H incident to $u(\mu)$ in pert(μ) appear between $f_{u(\mu)}$ and $l_{u(\mu)}$ in the counterclockwise order of the edges incident to $u(\mu)$ in $\mathcal{H}$. The first and last edge of $v(\mu)$ are defined analogously. Note that if the edges of H that belong to pert(μ) and are incident to $u(\mu)$ do not form an interval in the rotation of u in $\mathcal{H}$, then the skeleton of μ has no edge-compatible embedding and the Peg $(G, H, \mathcal{H})$ is nonplanar.

After a node μ of $\mathcal{T}$ has been visited by the algorithm, edges $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$ are associated with μ. We can then also refer to them as $f_{u(e)}$, $l_{u(e)}$, $f_{v(e)}$, and $l_{v(e)}$, where e is the virtual edge corresponding to μ in the skeleton of the parent of μ.

If μ is a Q- or an S-node, no check is needed. As skel(μ) is a cycle (possibly of length 2 in case of a Q-node), the only planar embedding of skel(μ) is edge-compatible with $\mathcal{H}$. The edges $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$ are easily computed.

If μ is an R-node, then skel(μ) has only two planar embeddings. For each of them, verify if it is edge-compatible with $\mathcal{H}$ by performing the following check. For each vertex x of skel(μ), restrict the circular list of its incident virtual edges to the virtual edges $e_1, \dots, e_h$ that contain an edge of H incident to x. Check if $l_{x(e_i)}$ precedes $f_{x(e_{i+1})}$ (for $i = 1, \dots, h$, where $e_{h+1} = e_1$) in the list $E_H(x)$ of edges incident to x in $\mathcal{H}$. If x is a pole, do an analogous check on the linear list of its incident virtual edges obtained by removing the parent edge from the circular list. If one of the tests succeeds, then select the corresponding embedding for skel(μ). Set $f_{u(\mu)} = f_{u(f_1)}$, $l_{u(\mu)} = l_{u(f_p)}$, $f_{v(\mu)} = f_{v(g_1)}$, and $l_{v(\mu)} = l_{v(g_q)}$, where f_1 and f_p (g_1 and g_q) are the first and the last virtual edge in the linear list of the virtual edges containing an edge of H and incident to $u(\mu)$ (respectively to $v(\mu)$).

If μ is a P-node, an embedding of skel(μ) is a counterclockwise order of its virtual edges around $u(\mu)$. We describe how to verify whether an embedding of skel(μ) exists that is edge-compatible with $\mathcal{H}$.

Consider the virtual edges containing edges of H incident to $u(\mu)$. We show how to construct a list L_u of such edges corresponding to the ordering they have in any embedding of skel(μ) that is edge-compatible with $\mathcal{H}$. Insert one such edge, say e_i, into L_u. Repeatedly consider the last element e_j of L_u, and insert as the new last element of L_u the edge e_{j+1} such that $l_{u(e_j)}$ immediately precedes $f_{u(e_{j+1})}$ in the counterclockwise order of the edges incident to $u(\mu)$ in $\mathcal{H}$. If $e_{j+1} = e_i$, then L_u is the desired circular list. If e_{j+1} does not exist, then the edge following $l_{u(e_j)}$ belongs to the parent edge of μ. Then, consider the first edge e_i. Repeatedly consider the first element e_j of L_u, and insert as the new first element of L_u the edge e_{j-1} such that $f_{u(e_j)}$ immediately follows $l_{u(e_{j-1})}$ in the counterclockwise order of the edges incident to $u(\mu)$ in $\mathcal{H}$. If e_{j-1} does not exist, then check whether all virtual edges containing edges of H incident to $u(\mu)$ have been processed, and in this case insert the the parent edge of μ as the first element of L_u. Analogously, construct a list L_v.

Let L_{uv} be the sublist obtained by restricting L_u to those edges that appear in L_v. Let L_{vu} be the corresponding sublist of L_v. Check whether L_{uv} and L_{vu} are the reverse of each other. If this is the case, a list L of the virtual edges of skel(μ) containing edges of H incident to $u(\mu)$ or to $v(\mu)$ can easily be constructed compatible with both L_u and L_v.

Finally, arbitrarily insert into L the virtual edges of skel(μ) not in L_u and not in L_v, thus obtaining an embedding of skel(μ) edge-compatible with $\mathcal{H}$.

Denote by f_1 and f_p (by g_1 and g_q) the virtual edges containing edges of H incident to $u(\mu)$ (respectively to $v(\mu)$) following and preceding the parent edge of μ in L. Set $f_{u(\mu)} = f_{u(f_1)}$, $l_{u(\mu)} = l_{u(f_p)}$, $f_{v(\mu)} = f_{v(g_1)}$, and $l_{v(\mu)} = l_{v(g_q)}$.

THEOREM 4.2. *Let $(G, H, \mathcal{H})$ be an n-vertex instance of* PEP *such that G is biconnected and H is connected. Algorithm BC solves* PEP *for $(G, H, \mathcal{H})$ in $O(n)$ time.*

PROOF. We show that Algorithm BC processes each node μ of $\mathcal{T}$ in $O(k_\mu)$ time, where k_μ is the number of children of μ in $\mathcal{T}$.

First, observe that the computation of $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$ is trivially done in $O(1)$ time once the embedding of skel(μ) has been decided.

If μ is a Q-node or an S-node, Algorithm BC does not perform any check or embedding choice.

If μ is an R-node, Algorithm BC computes the two planar embeddings of skel(μ) in $O(k_\mu)$ time. For each of these embeddings, Algorithm BC processes each vertex x of skel(μ) separately, considering the list of the virtual edges incident to x (which is trivially constructed in $O(t)$ time, where t is the number of such edges), and restricting the list to those virtual edges containing an edge of H incident to x (for each virtual edge, it suffices to check whether the first edge incident to x is associated with an edge of H, which is done in $O(1)$ time). Checking whether $l_{x(e_i)}$ precedes $f_{x(e_{i+1})}$ in the list of the edges incident to x in $\mathcal{H}$ is done in $O(1)$ time. Hence, the total time spent for each node x is $O(t)$. Summing up over all nodes of skel(μ) results in a total $O(k_\mu)$ time, as every edge is incident to two nodes and the total number of edges in skel(μ) is $O(k_\mu)$.

If μ is a P-node, extracting the virtual edges of skel(μ) containing edges of H incident to $u(\mu)$ or to $v(\mu)$ can be done in $O(k_\mu)$ time, as in the R-node case. For each of such edges, equipping $f_{u(e)}$, $l_{u(e)}$, $f_{v(e)}$, and $l_{v(e)}$ with a link to e is done in constant time. Determining an ordering of the virtual edges containing edges of H incident to $u(\mu)$ can be done in $O(k_\mu)$ time, as the operations performed for each virtual edge e_i are accessing the first and the last edge of e_i, accessing the edge following the last edge of e_i (preceding the first edge of e_i) in the counterclockwise order of the edges incident to $u(\mu)$ in $\mathcal{H}$, and accessing a virtual edge linked from the first or last edge; each of these operations is trivially done in $O(1)$ time. Marking the virtual edges in L_u and in L_v is done in $O(k_\mu)$ time, as L_u and L_v have $O(k_\mu)$ elements. Then, obtaining L_{uv} and L_{vu}, and checking whether they are the reverse of each other, is done in $O(k_\mu)$ time. Finally, extending L_{uv} to L is also easily done in $O(k_\mu)$ time; namely, if L_{uv} is empty, then let L be the concatenation of L_u and L_v (where such lists are made linear by cutting them at any point). Otherwise, start from an edge e_i of L_{uv}; e_i is also in L_u and in L_v; insert e_i into L; insert into L all the edges of L_u following e_i until the next edge e_{i+1} of L_{uv} has been found; insert into L all edges of L_v preceding e_i until the next edge e_{i+1} of L_{uv} has been found; insert e_{i+1} into L, and repeat the procedure. Each element of L_{uv}, L_u, and L_v is visited once, and hence such a step is performed in $O(k_\mu)$ time.

As $\sum_{\mu \in \mathcal{T}} k_\mu = O(n)$, the total running time of the algorithm is $O(n)$. □

Note that although Algorithm BC relies only on the assumptions that G is biconnected and H is connected, we will only use it in the more special case where H is also biconnected.

4.2. *G* Biconnected, All Vertices and Edges of *G* Lie in the Same Face of $\mathcal{H}$

The PEGs considered in this section are denoted by $(G(f), H(f), \mathcal{H}(f))$. Such instances are assumed to satisfy the following properties: (i) $G(f)$ is biconnected, (ii) $G(f)$ and $H(f)$ have the same vertex set, (iii) all vertices and edges of $H(f)$ are incident to the same face f of $\mathcal{H}(f)$, and (iv) no edge of $G(f)\backslash H(f)$ connects two vertices of the same block of $H(f)$. Algorithm BF, which deals with such a setting, is used as a subroutine by Algorithm BA, to be shown later, dealing with the instances of PEP in which G is biconnected and H is arbitrary.

First, we show that the structure of the cycles in $H(f)$ is very special.

PROPERTY 3. *Every simple path with at least two vertices of $H(f)$ is contained in at most one simple cycle of $H(f)$.*

PROOF. Suppose that there exists a path (that can possibly be a single edge) of $H(f)$ belonging to at least two simple cycles of $H(f)$. Then, such cycles share edges and define at least three regions of the plane. Not all edges of the two cycles can be incident to the same region, contradicting the fact that all edges of $H(f)$ are incident to the same region of the plane in $\mathcal{H}(f)$. □

Since all vertices and edges are incident to f, the only relevant cycles for which cycle–vertex constraints have to be checked are the facial cycles of f. We exploit this particular structure of the input to simplify the test of cycle compatibility with $\mathcal{H}(f)$ for the skeleton of a node μ of $\mathcal{T}(f)$, where $\mathcal{T}(f)$ is the SPQR-tree of $G(f)$.

LEMMA 4.3. *Consider any node μ of $\mathcal{T}(f)$. Then, an embedding of* skel(μ) *is cycle-compatible with $\mathcal{H}(f)$ if and only if for every facial cycle $\vec{C}$ of $\mathcal{H}(f)$ whose edges project to a cycle $\vec{C}'$ of* skel(μ)*, no vertex and no edge of* skel(μ) *is to the right of $\vec{C}'$, where $\vec{C}'$ is oriented according to the orientation of $\vec{C}$.*

PROOF. By assumption (iii) on the input, all vertices and edges of $H(f)$ are incident to the same face f of $\mathcal{H}(f)$. By construction, every facial cycle $\vec{C}$ of $H(f)$ is oriented in such a way that f and hence all vertices of $H(f)$ are to the left of $\vec{C}$. Then, by Lemma 3.4, if the edges of $\vec{C}$ determine a cycle $\vec{C}'$ of virtual edges of skel(μ), all vertices of skel(μ) that are not in $\vec{C}$ and all virtual edges of skel(μ) that are not in $\vec{C}'$ and that contain vertices of $G(f)$ have to be to the left of $\vec{C}'$. Finally, all virtual edges that are not in $\vec{C}'$ and that do not contain any vertex of $G(f)$ (i.e., virtual edges corresponding to Q-nodes) have one end-vertex that is not in $\vec{C}$, by assumption (iv) on the input. Such an end-vertex forces the edge to be to the left of $\vec{C}'$. □

To find compatible embeddings for the skeletons of the nodes of $\mathcal{T}(f)$, we again need to find edge-compatible embeddings, which can be done as in Algorithm BC. However, unlike Algorithm BC, Algorithm BF has also to make embedding choices to satisfy cycle compatibility constraints. Such constraints are the ones expressed by Lemma 4.3: for any node μ of $\mathcal{T}(f)$, the sought-after embedding of skel(μ) is such that for every facial cycle $\vec{C}$ of $\mathcal{H}(f)$ whose edges project to a cycle $\vec{C}'$ of skel(μ), no vertex and no edge of skel(μ) is to the right of $\vec{C}'$, where $\vec{C}'$ is oriented according to the orientation of $\vec{C}$.

The choice of an embedding for skel(μ) does not affect whether the cycle compatibility constraints are satisfied for the facial cycles $\vec{C}$ of $\mathcal{H}(f)$ whose edges project to a single virtual edge of skel(μ). On the other hand, the choice of an embedding for skel(μ) does affect whether the cycle compatibility constraints are satisfied for the facial cycles $\vec{C}$ of $\mathcal{H}(f)$ whose edges project to a cycle $\vec{C}'$ of skel(μ). Hence, to ensure cycle compatibility, we need to (quickly) find the projections of the facial cycles of $\mathcal{H}(f)$ in the skeletons of the nodes of $\mathcal{T}(f)$ and need to perform embedding choices for such skeletons that satisfy the constraints of Lemma 4.3. These tasks are simplified by the following two observations.

First, the facial cycles $\vec{C}$ of $\mathcal{H}(f)$ whose edges project to a cycle $\vec{C}'$ of skel(μ) are composed of a sequence of *traversing paths* for the neighbors of μ in $\mathcal{T}(f)$: for any neighbor ν of μ in $\mathcal{T}(f)$, a traversing path is a path between $u(\nu)$ and $v(\nu)$ that is composed of edges of $H(f)$, that belongs to pert(ν), and that is part of a facial cycle $\vec{C}$ of $\mathcal{H}(f)$ not entirely contained in pert(ν).

Second, by Property 3, every edge of $H(f)$ (and hence every path of $H(f)$) can be contained in at most one facial cycle of $\mathcal{H}(f)$. Therefore, a single flag suffices to encode the existence of a traversing path for a node of $\mathcal{T}(f)$.

We now give a high-level description of Algorithm BF.

Like Algorithm BC, Algorithm BF starts with the construction of the SPQR-tree $\mathcal{T}(f)$ of $G(f)$, roots it at an arbitrary Q-node, and visits $\mathcal{T}(f)$ in bottom-up order in such a way that after a node μ of $\mathcal{T}(f)$ has been visited, an embedding of skel(μ) that is compatible with $\mathcal{H}(f)$ is selected, if it exists.

To deal with edge compatibility constraints, Algorithm BF maintains edges $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$ for each node μ of $\mathcal{T}(f)$ (and for each virtual edge in the skeleton of each node of $\mathcal{T}(f)$) as in Algorithm BC. Additionally, to deal with cycle compatibility constraints, the algorithm maintains a flag $p(\mu)$ for each node μ of $\mathcal{T}(f)$ such that $p(\mu)$ is set to True if there exists a traversing path P for μ; flag $p(\mu)$ is set to False otherwise. Furthermore, to encode the direction of P the algorithm maintains a flag $uv(\mu)$. If $p(\mu) =$ True, the flag $uv(\mu)$ is set to True if P is oriented from $u(\mu)$ to $v(\mu)$ according to the orientation of $\vec{C}$, and it is set equal to False otherwise. We also refer to these flags as $p(e)$ and $uv(e)$, where e is the virtual edge corresponding to μ in the skeleton of the parent of μ.

Now, we state lemmas specifically dealing with S-, R-, and P-nodes of $\mathcal{T}(f)$. These lemmas will be used in the description of Algorithm BF.

Lemma 4.4. *Let μ be an S-node of $\mathcal{T}(f)$ with children $\mu_1, \mu_2, \ldots, \mu_k$. Then, $p(\mu_i) =$ true for some $1 \leq i \leq k$ if and only if $p(\mu_j) =$ true for all $1 \leq j \leq k$.*

Proof. If $p(\mu_j) = \text{TRUE}$ for all $1 \le j \le k$, then trivially $p(\mu_i) = \text{TRUE}$. If $p(\mu_i) = \text{TRUE}$ for some $1 \le i \le k$, there exists a traversing path of μ_i that is part of a simple cycle $\vec{C}$ of $H(f)$ not entirely contained in pert(μ_i); however, as μ is an S-node, $\vec{C}$ does not entirely lie inside pert(μ), as otherwise it would entirely lie inside pert(μ_i). Then, $\vec{C}$ consists of a traversing path of pert(μ_j), for all $1 \le j \le k$, and of a traversing path of the parent edge of skel(μ), thus proving the lemma. □

Next, we derive a simple criterion for an embedding of an R-node to be cycle-compatible. By Lemma 4.3, an embedding of a skeleton skel(μ) is cycle-compatible if for each facial cycle $\vec{C}$ of f that projects to a cycle $\vec{C}'$ in skel(μ), the right side of $\vec{C}'$ is empty. For an R-node μ this condition can be reformulated: each such cycle $\vec{C}'$ must have a face on its right side.

Lemma 4.5. *Let μ be an R-node of $\mathcal{T}(f)$. If an edge e of* skel(μ) *has a traversing path belonging to a facial cycle $\vec{C}$, let us orient e in the direction determined by the projection of $\vec{C}$ in* skel(μ)*. An embedding of* skel(μ) *is cycle-compatible with $\mathcal{H}(f)$ if and only if for each face g of the embedding of* skel(μ)*, either (i) every virtual edge e on the boundary of g is oriented so that g is to the right of e or (ii) none of the virtual edges on the boundary of g is oriented in a way that g is to the right of it.*

Proof. Suppose that an embedding of skel(μ) is cycle-compatible with $\mathcal{H}(f)$. Let g be a face of the embedding. If no edge e on the boundary of g contains a traversing path, then g satisfies condition (ii). Otherwise, assume that on the boundary of g there is an edge e containing a traversing path P such that g is to the right of e. Let $\vec{C}$ be the facial cycle of $\mathcal{H}(f)$ that contains P. By Lemma 4.3, $\vec{C}$ projects to a directed cycle $\vec{C}'$ in skel(μ), and no vertex or edge of skel(μ) is embedded to the right of $\vec{C}'$. Thus, $\vec{C}'$ corresponds to the boundary of the face g, and hence g satisfies condition (i).

Suppose now that in an embedding of skel(μ), every face satisfies condition (i) or condition (ii). We claim that the embedding of skel(μ) is cycle-compatible with $\mathcal{H}(f)$. To prove it, we use Lemma 4.3. Let $\vec{C}$ be a facial cycle of $\mathcal{H}(f)$ that projects to a simple cycle $\vec{C}'$ in skel(μ). Let e be any edge of $\vec{C}'$ and let g be the face to the right of e in the embedding of skel(μ). Necessarily, g satisfies condition (i). Hence, each edge on the boundary of g has a traversing path. The union of these paths forms a cycle in $\mathcal{H}(f)$, and by Property 3, this cycle is equal to $\vec{C}$. Thus, the boundary of g coincides with the cycle $\vec{C}'$. In particular, no vertex and no edge of skel(μ) is embedded to the right of $\vec{C}'$. By Lemma 4.3, this means that the embedding of skel(μ) is cycle-compatible with $\mathcal{H}(f)$. □

We next deal with P-nodes. The special structure of the Pegs $(G(f), H(f), \mathcal{H}(f))$ considered in this section implies that for each P-node μ, there exists at most one facial cycle $\vec{C}$ of f that projects to a cycle in skel(μ).

Lemma 4.6. *Let μ be a P-node of $\mathcal{T}(f)$. There exist either zero or two virtual edges of* skel(μ) *containing a traversing path.*

Proof. If there exists one virtual edge e_i of skel(μ) containing a traversing path that is part of a simple cycle $\vec{C}$ of $H(f)$ not entirely contained in pert(e_i), another virtual edge of skel(μ) containing a traversing path that is part of $\vec{C}$ exists, as otherwise $\vec{C}$ would not be a cycle. Further, if there exist at least three virtual edges of skel(μ) containing traversing paths, then each such path belongs to two simple cycles, thus contradicting Property 3. □

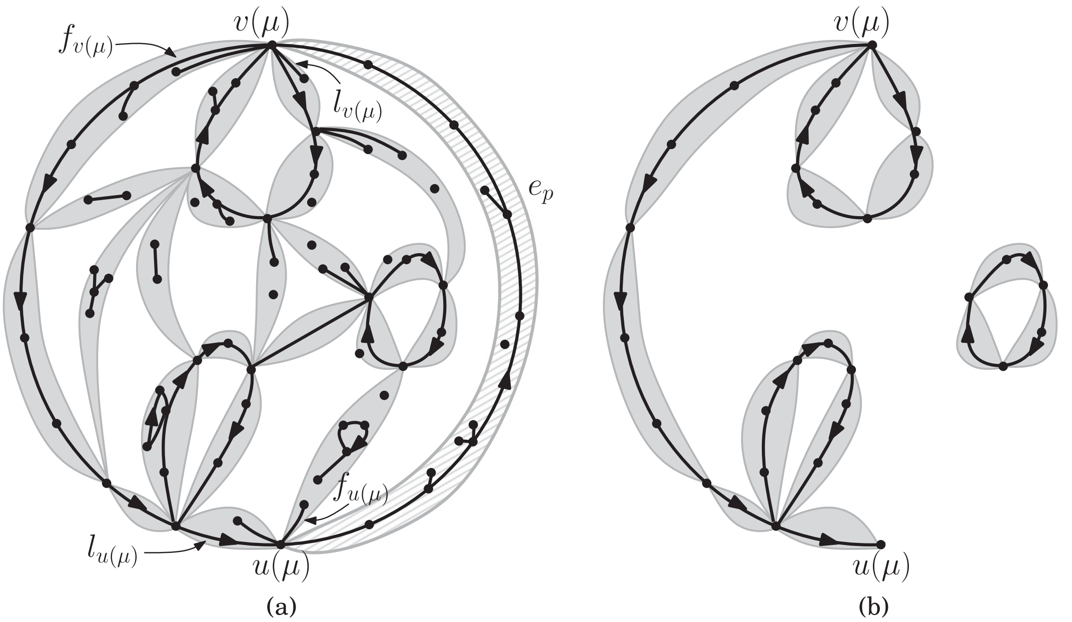


Fig. 9. Illustration for the case in which μ is an R-node. (a) One of the two embeddings of skel(μ). Grey regions represent expansion graphs of the virtual edges of the skeleton of μ. The expansion graph of the parent edge e_p is shaded. Edges in $G(f)$ not in $H(f)$ are not shown. (b) Graph skel$'(\mu)$ obtained by restricting skel(μ) to those edges $e_i \neq e_p$ with $p(e_i) = $ TRUE. The only edges of $H(f)$ shown are those belonging to traversing paths for children of μ in $\mathcal{T}(f)$.

Hence, the skeleton of every P-node contains at most one such cycle, whose right side must be empty by Lemma 4.3. This considerably simplifies the problem of finding a cycle-compatible embedding of a P-node. We are now ready to exhibit the main steps of Algorithm BF.

Algorithm BF. As stated earlier, Algorithm BF performs a bottom-up traversal of the rooted SPQR-tree $\mathcal{T}(f)$ of $G(f)$ such that for each processed node μ, a compatible embedding of skel(μ) is computed, if it exists. The algorithm computes the edges $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$ for each node μ of $\mathcal{T}(f)$ (and for each virtual edge in the skeleton of each node of $\mathcal{T}(f)$) as in Algorithm BC to find edge-compatible embeddings. Further, it computes the flags $p(\mu)$ and $uv(\mu)$ for each processed node to identify facial cycles of f that project to cycles in skel(μ). We now give a detailed description of how Algorithm BF processes a node μ, assuming that all flags and edges for all children of μ have already been computed.

If μ is a Q- or an S-node, no check is needed. As skel(μ) is a cycle, the only planar embedding of skel(μ) is compatible with $\mathcal{H}(f)$. Edges $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$, as well as flags $p(\mu)$ and $uv(\mu)$, can easily be computed. In particular, by Lemma 4.4, if μ is an S-node, then $p(\mu) = p(\mu_i)$ for any child μ_i of μ.

If μ is an R-node, as in Figure 9(a), then for each of the two planar embeddings of skel(μ), check if it is edge-compatible with $\mathcal{H}(f)$ and set values for $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$ as in Algorithm BC. To check if any of the two embeddings is cycle-compatible with $\mathcal{H}(f)$, we check if the conditions of Lemma 4.5 are satisfied. To perform this test, we need, for each virtual edge $e = uv$ of skel(μ), the corresponding flags $p(e)$ and $uv(e)$. This information is already known for all virtual edges, except the parent edge e_p of skel(μ). To compute the flags for e_p, we need to determine whether the virtual edge e_p contains a traversing path P_p and, if it does, determine its orientation. By definition of traversing path, P_p exists if and only if there exists a traversing path in pert(μ). Restrict skel(μ) to those edges $e_i \neq e_p$ with $p(e_i) = $ TRUE, and denote by skel$'(\mu)$ the

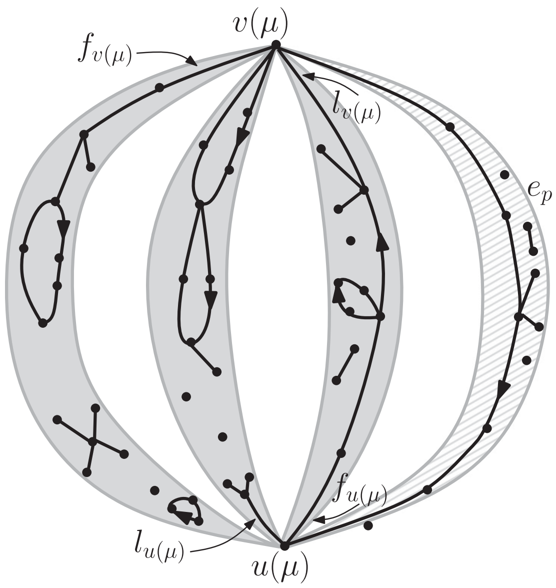


Fig. 10. Illustration for the case in which μ is a P-node. An embedding of skel(μ) is shown. Grey regions represent expansion graphs of the virtual edges of the skeleton of μ. The expansion graph of the parent edge e_p is shaded. Edges in $G(f)$ not in $H(f)$ are not shown.

resulting graph. Refer to Figure 9(b). Note that by Property 3, for each virtual edge $e_i \in$ skel$'(\mu)$, there exists exactly one traversing path in pert(e_i), and this traversing path is contained in exactly one simple cycle of $H(f)$. In addition, for each simple cycle $\vec{C}$ of $H(f)$ passing through a vertex of skel$'(\mu)$, there exist exactly two virtual edges of skel$'(\mu)$ incident to this vertex that contain a traversing path that is part of $\vec{C}$. These two observations imply that a traversing path P_p exists if and only if both the degree of $u(\mu)$ and the degree of $v(\mu)$ in skel$'(\mu)$ are odd. Hence, if both the degree of $u(\mu)$ and the degree of $v(\mu)$ are odd, then set $p(\mu) =$ TRUE and $p(e_p) =$ TRUE; otherwise, set $p(\mu) =$ FALSE and $p(e_p) =$ FALSE. In the former case, the orientation of P_p is the only one that makes the number of edges e_i incident to $u(\mu)$ with $uv(e_i) =$ TRUE equal to the number of edges e_i incident to $u(\mu)$ with $uv(e_i) =$ FALSE; this determines $uv(\mu)$ and $uv(e_p)$.

Now, $p(e_i)$ and $uv(e_i)$ are defined for every virtual edge e_i of skel(μ). Consider every face g of skel(μ) and denote by $e_j = (u_j, v_j)$ any edge incident to g. Suppose, without loss of generality, that g is to the right of e_j when traversing it from u_j to v_j. Then, check if $p(e_j) =$ FALSE, or $p(e_j) =$ TRUE and $uv(e_j) =$ FALSE for all edges e_j incident to g, and check whether $p(e_j) =$ TRUE and $uv(e_j) =$ TRUE for all edges e_j incident to g. If one of the two checks succeeds, the face does not violate Lemma 4.5, but otherwise it does.

If μ is a P-node, as in Figure 10, check if an embedding of skel(μ) exists that is compatible with $\mathcal{H}(f)$ as follows. By Lemma 4.6, there exist either zero or two virtual edges of skel(μ) containing a traversing path. Then, consider the children μ_i of μ such that $p(\mu_i) =$ TRUE. If zero or two such children exist, then the parent edge of skel(μ) has no traversing path; if one such a child exists, then the parent edge of skel(μ) has a traversing path. Denote by e_i and e_j the edges of skel(μ) containing a traversing path, if such edges exist, where possibly e_j is the parent edge (in this case, set $p(e_j) =$ TRUE, and set $uv(e_j) =$ TRUE if $uv(e_i) =$ FALSE and $uv(e_j) =$ FALSE otherwise). If there exists no edge e_i of skel(μ) such that $p(e_i) =$ TRUE, then construct an embedding of skel(μ) that is edge-compatible with $\mathcal{H}(f)$, if possible, as in Algorithm BC; as there exists no facial cycle of $\mathcal{H}(f)$ whose edges belong to distinct virtual edges of skel(μ), then an edge-compatible embedding is also cycle-compatible with $\mathcal{H}(f)$. Edges $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$ are computed as in Algorithm BC. Set flag $p(\mu) = $ FALSE. If there exist two edges e_i and e_j such that $p(e_i) = $ TRUE, $p(e_j) = $ TRUE, and $p(e_l) = $ FALSE for every edge $e_l \neq e_i, e_j$, suppose that $uv(e_i) = $ TRUE and $uv(e_j) = $ FALSE, the case in which $uv(e_i) = $ FALSE and $uv(e_j) = $ TRUE being analogous. Note that the expansion graphs of e_i and e_j must contain at least one H-edge incident to $u(\mu)$ as well as at least one H-edge incident to $v(\mu)$. By Lemma 4.3, e_j has to immediately precede e_i in the counterclockwise order of the edges incident to $u(\mu)$. Then, construct L_u and L_v as in Algorithm BC; check whether L_u and L_v, restricted to the edges that appear in both lists, are the reverse of each other; and further, check whether e_j precedes e_i in L_u and whether e_i precedes e_j in L_v. If the checks are positive, construct the list L of all edges of skel(μ) as in Algorithm BC, except for the fact that the edges of skel(μ) not in L_u and not in L_v are not inserted between e_j and e_i. Edges $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$ are computed as in Algorithm BC. Set $p(\mu) = $ FALSE if e_j corresponds to a child μ_j of μ and $p(\mu) = $ TRUE if e_j is the parent edge of μ; in the latter case, $uv(\mu) = $ TRUE if $uv(\mu_i) = $ TRUE and $uv(\mu) = $ FALSE otherwise.

We get the following theorem.

THEOREM 4.7. *Let $(G(f), H(f), \mathcal{H}(f))$ be a biconnected* PEG *with n vertices such that $G(f)$ and $H(f)$ have the same vertex set, all vertices and edges of $H(f)$ are incident to the same face f of $\mathcal{H}(f)$, and no edge of $G(f)\setminus H(f)$ connects two vertices belonging to the same block of $H(f)$. Algorithm BF solves* PEP *for $(G(f), H(f), \mathcal{H}(f))$ in $O(n)$ time.*

PROOF. We show that Algorithm BF processes each node μ of $\mathcal{T}(f)$ in $O(k_\mu)$ time, where $\mu_1, \ldots, \mu_{k_\mu}$ are the children of μ in $\mathcal{T}(f)$.

Observe that the computation of $f_{u(\mu)}$, $l_{u(\mu)}$, $f_{v(\mu)}$, and $l_{v(\mu)}$ and the check of edge-compatibility are done as in Algorithm BC; hence, they take $O(k_\mu)$ time. We describe how to check the cycle compatibility of an embedding of skel(μ) in $O(k_\mu)$ time.

If μ is a Q-node or an S-node, Algorithm BF neither performs any checks nor does it make any embedding choices.

If μ is a P-node, then Algorithm BF performs the same checks and embedding choices as Algorithm BC, plus the check that the two edges e_i and e_j with $p(e_i) = $ TRUE and $p(e_j) = $ FALSE (notice that one of these edges could be the the parent edge of μ) are consecutive (with the right order) in L_u and L_v. This is done in constant time. Flags $p(\mu)$ and $uv(\mu)$ are computed in $O(k_\mu)$ time, by simply checking the flags $p(\mu_i)$ and $uv(\mu_i)$, for $i = 1, \ldots, k$.

Suppose that μ is an R-node. The construction of skel$'(\mu)$ can easily be done in $O(k_\mu)$ time, as such a graph can be obtained from skel(μ) by simply checking flag $p(e_i)$, for each edge e_i in skel(μ). Then, the degree of $u(\mu)$ and $v(\mu)$ in skel$'(\mu)$, as well as the flags $p(\mu)$, $uv(\mu)$, $p(e_p)$ and $uv(e_p)$, can be computed in total $O(k_\mu)$ time. The test on each face takes time linear in the number of edges incident to the face. Namely, such a test consists of two checks, each of which requires considering a constant number of flags associated with each edge of the face. As every edge is incident to two faces of skel(μ) and the number of edges in skel(μ) is $O(k_\mu)$, the total time spent for the test on the faces of skel(μ) is $O(k_\mu)$.

As $\sum_{\mu \in \mathcal{T}} k_\mu = O(n)$, the total running time of the algorithm is $O(n)$. □

4.3. *G* Biconnected

In this section, we show how to solve PEP for general biconnected PEGS—that is PEGS $(G, H, \mathcal{H})$ where G is biconnected and H is arbitrary. The algorithm employs the algorithms from the previous two sections as subroutines. The general outline is as follows. First, compute a subgraph H^+ of G with the following properties: (i) H^+ is biconnected; (ii) H is a subgraph of H^+; and (iii) H^+ contains every nonlocal H-bridge of G. Second, solve instance $(H^+, H, \mathcal{H})$ obtaining an embedding $\mathcal{H}^+$ of H^+ extending $\mathcal{H}$, if H^+ admits one. We will show that this step can be reduced to several applications of Algorithm BF. Finally, solve instance $(G, H^+, \mathcal{H}^+)$ with Algorithm BC—we will see that H^+ is connected (even biconnected), and hence the algorithm can be applied.

In a first step, we ensure that all nonlocal H-bridges of G are trivial. Recall that every nontrivial nonlocal H-bridge K has at most one candidate face f_K of $\mathcal{H}$ where it can be embedded. We obtain the graph H' with embedding $\mathcal{H}'$ as in Definition 3.7 by adding the vertices of $K - V(H)$ to H and embedding them into the face f_K for each nontrivial nonlocal H-bridge K of G.

Let H^+ be the graph obtained from G by removing the vertices and edges (but not the attachments) of all local H-bridges of G. Note that $H' \subseteq H^+$, that H^+ and H' have the same vertex set, and that any embedding of H^+ that extends $\mathcal{H}$ also extends $\mathcal{H}'$ and vice versa.

Each H'-bridge K of H^+ is nonlocal, and therefore there exists a unique face f_K where it needs to be embedded. Since H-bridges that are embedded in distinct faces of $\mathcal{H}$ do not interact, we can solve the instances stemming from the faces of $\mathcal{H}$ independently, which enables us to use Algorithm BF to find an embedding extension of H^+. This motivates the following definitions, which take a more local view at the PEG $(H^+, H', \mathcal{H}')$. Let f be a face of $\mathcal{H}'$ and let $V(f)$ be the set of vertices of H' that are incident to f. Let $H(f)$ be the subgraph of H' induced by $V(f)$, let $\mathcal{H}(f)$ be $\mathcal{H}'$ restricted to $H(f)$, and let $G(f)$ be the subgraph of H^+ induced by $V(f)$. By construction, in any embedding of H^+ that extends $\mathcal{H}$, the edges of $G(f)$ not belonging to $H(f)$ are embedded inside f.

Our approach is to first find an embedding $\mathcal{H}^+$ of H^+ that extends $\mathcal{H}'$ (i.e., solve PEP for $(H^+, H', \mathcal{H}')$) and then find an embedding $\mathcal{G}$ for $(G, H^+, \mathcal{H}^+)$ (i.e., solve PEP for $(G, H^+, \mathcal{H}^+)$). The latter step is actually simple, as H^+ is biconnected and thus connected. Therefore, Algorithm BC can be used to solve this subproblem.

LEMMA 4.8. *H^+ is biconnected.*

PROOF. By construction of H^+, each H^+-bridge of G has all of its attachment vertices in the same block of H, and hence in the same block of H^+, as H is a subgraph of H^+. Therefore, the number of blocks of H^+ is not modified by the addition of the H^+-bridges of G. Since such an addition produces G, which is biconnected, it follows that H^+ is biconnected. □

Clearly, if $(G, H, \mathcal{H})$ is planar, then an embedding $\mathcal{G}$ of G extending $\mathcal{H}$ exists, and the restriction of $\mathcal{G}$ to H^+ yields an intermediate embedding $\mathcal{H}^+$ of H^+ extending $\mathcal{H}'$, which can then be extended to an embedding of G extending $\mathcal{H}$. We show that the choice of $\mathcal{H}^+$ does not change the possibility of finding such an embedding extension. In particular, if $\mathcal{H}_1^+$ and $\mathcal{H}_2^+$ are two embeddings of H^+ extending $\mathcal{H}$, then the PEG $(G, H^+, \mathcal{H}_1^+)$ is planar if and only if $(G, H^+, \mathcal{H}_2^+)$ is planar.

LEMMA 4.9. *A biconnected* PEG $(G, H, \mathcal{H})$ *is planar if and only if (a) H^+ admits a planar embedding extending $\mathcal{H}$ and (b) for every planar embedding $\mathcal{H}^+$ of H^+, $(G, H^+, \mathcal{H}^+)$ is planar.*

PROOF. Clearly, if conditions (a) and (b) hold, then G has an embedding extending $\mathcal{H}$. To prove the converse, assume that G has an embedding $\mathcal{G}$ extending $\mathcal{H}$. Clearly, $\mathcal{G}$ contains a subembedding $\mathcal{H}^+$ of H^+ that extends $\mathcal{H}$, so condition (a) holds. It remains to prove that condition (b) holds as well.

First, we introduce some terminology. Let f be any face of $\mathcal{H}$ and let $\mathcal{H}^+$ be any embedding of H^+ that extends $\mathcal{H}$. In $\mathcal{H}^+$, the face f can be partitioned (by the edges of H^+ not in H) into several faces, which we will call the *subfaces of f*. A set of vertices $S \subseteq V(H)$ is said to be *mutually visible in f with respect to $\mathcal{H}^+$* if $\mathcal{H}^+$ has a subface of f that contains all vertices of S on its boundary.

The proof that condition (b) holds is based on two claims. The first one shows that for the vertices that belong to the same block of H, mutual visibility is independent of the choice of $\mathcal{H}^+$.

Claim 3. Let $\vec{C}$ be a facial cycle of f and let $S \subseteq V(\vec{C})$ be a set of vertices of $\vec{C}$. If the vertices in S are mutually visible in f with respect to at least one embedding of H^+ that extends $\mathcal{H}$, then they are mutually visible in f with respect to every embedding of H^+ that extends $\mathcal{H}$.

Note that the mutual visibility of S in f only depends on the embedding $\mathcal{H}^+$ restricted to $G(f)$. Let $\mathcal{T}$ be the SPQR-tree of $G(f)$. By Theorem 3.6, the embeddings of $G(f)$ that extend $\mathcal{H}(f)$ are exactly obtained by specifying a compatible embedding for the skeleton of each node of $\mathcal{T}$. Assume that $\mathcal{G}_1$ and $\mathcal{G}_2$ are two embeddings of $G(f)$ that extend $\mathcal{H}$. Assume that the vertices of S are mutually visible in f with respect to $\mathcal{G}_1$. We will show that they are also mutually visible with respect to $\mathcal{G}_2$. In view of Theorem 3.6, we may assume that $\mathcal{G}_2$ was obtained from $\mathcal{G}_1$ by changing the embedding of the skeleton of a single node $\mu \in \mathcal{T}$.

Let us distinguish two cases, depending on whether $\vec{C}$ is contained in the expansion graph of a single virtual edge of skel(μ) or whether it projects to a cycle in skel(μ).

If $\vec{C}$ is part of the expansion graph of a single virtual edge $e = \{x, y\} \in$ skel(μ), then let $\mathcal{G}_e$ be the embedded graph obtained as the union of the expansion graph of e and a single edge connecting x and y, embedded in the outer face of the expansion graph. We easily see that the vertices in S are mutually visible in f if and only if they share the same face of $\mathcal{G}_e$, other than the face that is to the right of $\vec{C}$. Since $\mathcal{G}_e$ does not depend on the embedding of skel(μ), the vertices in S are mutually visible in $\mathcal{G}_2$.

Assume now that the cycle $\vec{C}$ projects to a cycle $\vec{C}'$ in skel(μ). By Lemma 4.3, in any compatible embedding of skel(μ), all vertices and edges of skel(μ) that do not belong to $\vec{C}'$ are embedded to the left of $\vec{C}'$. In particular, if μ is an R-node, then skel(μ) only has a single compatible embedding. Thus, μ must be a P-node. Let e and e' be the two virtual edges of skel(μ) that form $\vec{C}'$. In each compatible embedding of skel(μ), these two edges must be embedded next to each other and in the same order. It follows easily that any two compatible embeddings of skel(μ) yield embeddings of $G(f)$ in which the vertices from S have the same mutual visibility. This completes the proof of the claim.

Let us proceed with the proof that condition (b) holds. We need more terminology. Let K and K' be a pair of local H-bridges of G whose attachments all appear on a facial cycle $\vec{C}$ of a face f in $\mathcal{H}$. We say that K and K' have a *three-vertex conflict on* $\vec{C}$ if they share at least three attachments, and that they have a *four-vertex conflict on* $\vec{C}$ if there are four vertices x, x', y, y' that appear on $\vec{C}$ in this cyclic order, and x, y are attachments of K, whereas x', y' are attachments of K'.

Claim 4. Assume that a face f_K of $\mathcal{H}$ has been assigned to every local H-bridge K of G so that all attachments of K are on the boundary of f_K. Let $\mathcal{H}^+$ be an embedding of H^+ extending $\mathcal{H}$. There is an embedding $\mathcal{G}$ of G extending $\mathcal{H}^+$, with the additional property that each local H-bridge K is embedded inside a subface of f_K if and only if:

(1) For any local H-bridge K, all attachments of K are mutually visible in f_K with respect to $\mathcal{H}^+$.
(2) If K and L are distinct local H-bridges assigned to the same face $f_K = f_L$ such that the attachments of K and L appear on a common facial cycle $\vec{C}$ of $\mathcal{H}^+$, then K and L have no conflict on $\vec{C}$.

Clearly, the two conditions are necessary. To prove that they are also sufficient, assume that both the conditions hold. Construct an embedding of $\mathcal{G}$ with the desired properties as follows. Let f be any face of $\mathcal{H}$. Observe that the first condition of the claim guarantees that for every H-bridge K assigned to f, there is a face f' of $\mathcal{H}^+$ that is a subface of f such that all attachments of K appear on the boundary of f'. Let f' be a face of $\mathcal{H}^+$ that is a subface of f, and let $K_1, \dots K_s$ be all local H-bridges that were assigned to f and whose attachments all appear on the boundary of f'. We show that all bridges $K_1, \dots, K_s$ can be embedded inside f'.

First, observe that the boundary of f' is a simple cycle C', because H^+ is biconnected. In addition, observe that no two bridges K_i and K_j have a conflict on C', by the second condition of the claim. To show that all bridges $K_1, \dots, K_s$ can be embedded inside C', proceed by induction on s. If $s = 1$, the statement is clear. Assume that $s \geq 2$ and that the bridge K_1 has been successfully embedded into f'. The embedding of K_1 partitions f' into several subfaces $f'_1, \dots, f'_t$. Such subfaces are again bounded by simple cycles; otherwise, G would not be biconnected. We claim that for every bridge K_i, with $i \geq 2$, there is a subface f'_j containing all attachments of K_i. Consider any bridge K_i. Assume first that K_i has an attachment x that is not an attachment of K_1. Then, x belongs to a unique subface f'_j. Hence, if K_i has another attachment not belonging to f'_j, there is a four-vertex conflict of K_1 and K_i on $\vec{C}'$, contradicting the second condition of the claim. Assume next that each attachment of K_i is also an attachment of K_1. Then, K_i has exactly two attachments, and if such attachments do not share a face f'_j, a four-vertex conflict of K_1 and K_i on $\vec{C}'$ is created, again contradicting the second condition of the claim.

We can thus assign to each K_i a subface f'_j that contains all of its attachments. By induction, all K_i's can be embedded into their assigned faces, thus proving the second claim.

The proof that condition (b) holds follows easily from the two claims. Namely, assume that G has an embedding $\mathcal{G}$ extending $\mathcal{H}$. Let $\mathcal{H}^+$ be $\mathcal{G}$ restricted to H^+. For every local H-bridge K of G, let f_K be the face of $\mathcal{H}$ inside which K is embedded in $\mathcal{G}$. Clearly, $\mathcal{H}^+$ satisfies the two conditions of the second claim, as it can be extended into $\mathcal{G}$. Then, every embedding of H^+ that extends $\mathcal{H}$ satisfies the two conditions of the second claim: for the first condition, this is a consequence of the first claim, and for the second condition, this is obvious. We conclude that every embedding of H^+ that extends $\mathcal{H}$ can be extended into an embedding of G, thus proving condition (b) and hence the lemma. □

As stated earlier, each H'-bridge K of $(H^+, H', \mathcal{H}')$ is nonlocal, and we therefore know into which face f_K it needs to be embedded. Since H'-bridges that are embedded in different faces do not interact, we can solve the subinstance $(G(f), H(f), \mathcal{H}(f))$ arising from each face separately. Clearly, if one of the instances fails, then G does not have an embedding extension. If all instances admit embedding extensions, gluing them together yields an embedding $\mathcal{H}^+$ of H^+ extending $\mathcal{H}'$. The previous lemma then implies that $(G, H, \mathcal{H})$ is planar if and only if $(G, H^+, \mathcal{H}^+)$ is planar. We are now ready to describe Algorithm BA (for G Biconnected and H Arbitrary).

Algorithm BA. Starting from an instance $(G, H, \mathcal{H})$ of PEP, graphs $G(f)$ and $H(f)$, and embedding $\mathcal{H}(f)$, for every face f of $\mathcal{H}$, are computed as follows. For each H-bridge K of G, determine whether it is local to a block of H or not. In the former case, K is not associated to any face f of $\mathcal{H}$. In the latter case, we compute the unique face f of $\mathcal{H}$ in which K has to be embedded in any solution of instance $(G, H, \mathcal{H})$ of PEP, and we associate K with f.

These computations can be performed in linear time by applying Lemma 2.2. To do so, we have to construct the $\mathcal{CF}$-tree of $\mathcal{H}$, the $\mathcal{BF}$-tree of $\mathcal{H}$, the $\mathcal{VF}$-graph of $\mathcal{H}$, and the enriched block-cutvertex tree of each connected component of H. As shown in Section 2.3, this can be done in linear time.

Then, for each face f of $\mathcal{H}$, consider every H-bridge K associated with f. Add the vertices and the edges of K to $G(f)$, and add the vertices of K to $\mathcal{H}(f)$ inside f. Let $H^+ = \bigcup_{f \in H} G(f)$. For each face f of $\mathcal{H}$, call Algorithm BF with input $(G(f), H(f), \mathcal{H}(f))$. If Algorithm BF succeeds for every instance $(G(f), H(f), \mathcal{H}(f))$ (thus providing an embedding $\mathcal{H}^+(f)$ of $G(f)$ whose restriction to $H(f)$ is $\mathcal{H}(f)$), merge the embeddings $\mathcal{H}^+(f)$ of $G(f)$ into a planar embedding $\mathcal{H}^+$ of H^+. Finally, call Algorithm BC with $(G, H^+, \mathcal{H}^+)$.

THEOREM 4.10. *Let $(G, H, \mathcal{H})$ be an n-vertex instance of* PEP *such that G is biconnected. Algorithm BA solves* PEP *for $(G, H, \mathcal{H})$ in $O(n)$ time.*

PROOF. The correctness of the algorithm follows from Lemma 4.9.

By Lemma 2.2, determining whether an H-bridge K is local or not can be done in time linear in the size of K. Further, if K is nonlocal, the only face of $\mathcal{H}$ incident to all attachment vertices of K can be computed, if it exists, in time linear in the size of K. Then, the construction of graphs $G(f)$, $H(f)$, H^+ and of embeddings $\mathcal{H}(f)$ takes $O(n)$ time, as it only requires to perform the union of graphs that have total $O(n)$ edges.

By Theorem 4.7, Algorithm BF runs in time linear in the number of edges of $G(f)$, and hence all executions of Algorithm BF take a total $O(n)$ time. By Theorem 4.2, Algorithm BC runs in $O(n)$ time, and hence the total running time of Algorithm BA is $O(n)$. $\square$

This concludes the case of biconnected PEGS.

4.4. *G* Connected or Disconnected

In this section, we give an algorithm that decides the planarity of general PEGS. First, we deal with instances $(G, H, \mathcal{H})$ of PEP in which G is connected, every nontrivial H-bridge of G is local, and H is arbitrary. We show that the three conditions of Lemma 3.8 can be checked in linear time. The first condition can be checked in linear time by Theorem 4.10. The second and the third conditions can be checked in linear time by the following two lemmas.

LEMMA 4.11. *Let $(G, H, \mathcal{H})$ be a connected* PEG. *Let $G_1, \ldots, G_t$ be the blocks of G, and let H_i be the subgraph of H induced by the vertices of G_i. There is a linear-time algorithm that checks whether any two distinct graphs among $H_1, \ldots, H_t$ alternate around any vertex of $\mathcal{H}$.*

PROOF. Let us describe the algorithm that performs the required checks. We assume that every edge e of H has an associated label indicating the block of G that contains e. We also associate to each block two integer counters that will be used in the algorithm.

We now describe a procedure TEST(x), which, for a given vertex $x \in V(H)$, checks whether any two graphs H_i, H_j alternate around x. Let us use the term *x-edge* to refer to any edge of H incident to x, and let *x-block* refer to any block of G that contains at least one x-edge.

The procedure TEST(x) proceeds as follows. First, for every x-block G_i, it determines the number of x-edges in G_i and stores this in a counter associated with G_i. This is done by simply looking at every edge incident to x and incrementing the counter of the corresponding block. Next, TEST(x) visits all x-edges in the order determined by the rotation $\sigma_{\mathcal{H}}(x)$, starting at an arbitrary x-edge. For each x-block, it maintains in a counter the number of its x-edges that have been visited so far. An x-block is *active* if some but not all of its x-edges have already been visited.

The procedure TEST(x) also maintains a stack containing the active x-blocks. At the beginning of the procedure, the counters of visited edges of each x-block are set to zero and the stack is empty.

For every edge e that TEST(x) visits, it performs the following steps:

(1) Let G_i denote the block containing e. Increment the counter of visited x-edges of G_i.
(2) If no other edge of G_i has been visited so far, push G_i on the stack.
(3) If some x-edge of G_i has been visited before e, we know that G_i is currently somewhere on the stack. Check whether G_i is on the top of the stack. If the top of the stack contains an x-block G_j different from G_i, output that H_i and H_j alternate around x and stop.
(4) Check whether e is the last x-edge of G_i to be visited (comparing its counter of visited x-edges to the counter of total x-edges), and if it is, pop G_i from the stack. (Note that if G_i has only one x-edge, it is pushed and popped during the visit of this edge.)

If TEST(x) visits all x-edges without rejecting, it outputs that there is no alternation around x.

The procedure TEST(x) takes time proportional to the number of x-edges. Thus, we can call TEST(x) for all vertices $x \in V(H)$ in linear time to test whether there is any alternation in $\mathcal{H}$.

Let us now argue that the procedure TEST(x) is correct. Assume that TEST(x) outputs an alternation of H_i and H_j. This can only happen when G_j is on the top of the stack while an x-edge $e \in G_i$ is visited, and furthermore, e is not the first edge of G_i to be visited. It follows that the first edge of G_i was visited before the first edge of G_j, and G_j is still active when e is visited. This shows that H_i and H_j indeed alternate around x.

Conversely, assume that there is a pair of graphs H_i and H_j that alternate around x, and the alternation is witnessed by two pairs of x-edges $e, e' \in H_i$ and $f, f' \in H_j$. For contradiction, assume that TEST(x) outputs that there is no alternation. Without loss of generality, assume that at least one x-edge of H_i is visited before any x-edge of H_j, that e is visited before e', and that f is visited before f'. Thus, the four x-edges are visited in the order e, f, e', f'. When the procedure visits e', both G_i and G_j are active, and G_j is on the stack above G_i since we assumed that the first x-edge of G_i is visited before the first x-edge of G_j. This means that when TEST(x) visited e', G_i was not on the top of the stack and an alternation should have been reported.

This contradiction completes the proof of the lemma. □

The next lemma shows that the third condition of Lemma 3.8 can also be tested in linear time, assuming that the first and second conditions of the lemma hold.

LEMMA 4.12. *Let $(G, H, \mathcal{H})$ be a connected* PEG. *Let $G_1, \dots, G_t$ be the blocks of G, and let H_i be the subgraph of H induced by the vertices of G_i. Let $\mathcal{H}_i$ be $\mathcal{H}$ restricted to H_i. Assume that the following conditions hold:*

(1) *each nontrivial H-bridge of G is local,*
(2) *each G_i has an embedding that extends $\mathcal{H}_i$, and*
(3) *no two of the graphs $H_1, \dots, H_t$ alternate around any vertex of H.*

There is a linear-time algorithm that decides whether there exists a facial cycle $\vec{C}$ of $\mathcal{H}$ that separates a pair of vertices x and y of H such that x and y are connected by a path of G that has no vertex in common with $\vec{C}$.

PROOF. Let P be a path in G with end-vertices in H, and let $\vec{C}$ be a facial cycle of $\mathcal{H}$. If P and $\vec{C}$ are vertex-disjoint and the end-vertices of P are separated by $\vec{C}$, we say that P and $\vec{C}$ form a *PC-obstruction*. A PC-obstruction $(P, \vec{C})$ is called *minimal* if

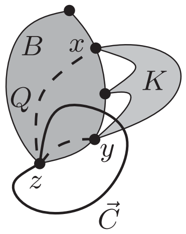


Fig. 11. Illustration for the case in which K is local to a block B of H. Two edges of the path $Q \subseteq B \subseteq G_i$ connecting x and y in H alternate with two edges of cycle $\vec{C} \in G_j$ around vertex z, which is a contradiction to condition 3 of the lemma.

no proper subpath $P' \subset P$ forms a PC-obstruction with $\vec{C}$. Observe that in a minimal PC-obstruction, all internal vertices of P belong to $V(G)\setminus V(H)$.

We want to show that the existence of a PC-obstruction can be tested in linear time. Of course, it is sufficient to test the existence of a minimal PC-obstruction. Before we explain how this test is done, we make some more observations concerning the structure of minimal PC-obstructions.

Let $(P, \vec{C})$ be a minimal PC-obstruction, and let x and y be the end-vertices of P. As the internal vertices of P belong to $V(G)\setminus V(H)$, P is a subgraph of an H-bridge K, and x and y are among the attachments of K. Let us now distinguish two cases, depending on whether K is local to some block or not.

First, assume that K is local to a block B of H. Refer to Figure 11. Then, both B and P are part of the same block G_i of G. Hence, $\vec{C}$ belongs to a different block of G, because if it belonged to G_i, then G_i would contain the whole PC-obstruction $(P, \vec{C})$ and it would be impossible to extend the embedding $\mathcal{H}_i$ to G_i, thus contradicting condition 2 of the lemma. Then, let G_j be the block of G that contains $\vec{C}$. Since x and y belong to a common block B of H, they are connected by a path $Q \subseteq B$. Since x and y are separated by $\vec{C}$, Q shares a vertex z with $\vec{C}$ (otherwise, the embedding $\mathcal{H}$ would not be planar). Since Q and $\vec{C}$ belong to distinct blocks, z is their unique common vertex. This, together with the fact that $\vec{C}$ separates x and y, implies that the two edges that belong to Q alternate with the two edges that belong to $\vec{C}$ in the rotation of z. Thus, G_i alternates with G_j around z, contradicting condition 3 of the lemma. Then, K cannot be a local bridge.

Suppose now that K is nonlocal. By condition 1 of the lemma, K consists of a single edge of $E(G)\setminus E(H)$. We conclude that any minimal PC-obstruction $(P, \vec{C})$ has the property that P is a single edge that forms a nonlocal H-bridge of G.

Observe that two vertices x and y belonging to distinct blocks of H are separated by a facial cycle of $\mathcal{H}$ if and only if there is no face of $\mathcal{H}$ to which both x and y are incident.

We are now ready to describe the algorithm that determines the existence of a minimal PC-obstruction. The algorithm tests all edges of $E(G)\setminus E(H)$ one by one. For any such edge e, it determines in constant time whether it is an H-bridge—that is, whether its endpoints x and y belong to H. If it is an H-bridge, it checks whether it is nonlocal in constant time by using Lemma 2.2. For a nonlocal bridge, the algorithm then checks in constant time whether there is a face f of H into which this bridge can be embedded, again using Lemma 2.2. Such a face f, if it exists, is uniquely determined, and its boundary contains x and y.

Overall, for any edge e, the algorithm determines in constant time whether this edge is a nonlocal bridge that is part of a minimal PC-obstruction. Hence, in linear time, we determine whether G has any PC-obstruction, thus concluding the proof of the lemma. $\square$

Combining Lemmas 2.2, 3.8, 4.11, and 4.12 with Theorem 4.10, we obtain the following result.

THEOREM 4.13. PEP *can be solved in linear time when restricted to instances* $(G, H, \mathcal{H})$ *where G is connected.*

PROOF. By Lemma 2.2, an instance of PEP where G is connected can be reduced in linear time to an equivalent instance that has the additional property that all nontrivial H-bridges are local. Namely, by Lemma 2.2, we may compute whether an H-bridge K is nonlocal and, in such case, which is the face of $\mathcal{H}$ in which K has to be embedded, in time linear in the size of K. We may thus assume that $(G, H, \mathcal{H})$ is an instance of PEP where G is simply connected and all nontrivial H-bridges in G are local to some block.

To solve PEP for $(G, H, \mathcal{H})$, we present an algorithm based on the characterization of Lemma 3.8. First, we generate all subinstances $(G_i, H_i, \mathcal{H}_i)$ for $i = 1, \dots, t$, induced by the blocks of G. It is not difficult to see that the subinstances can be generated in linear time. We then solve these subinstances using Algorithm BA, which takes linear time, by Theorem 4.10, since the total size of the subinstances is linear. If any of the subinstances does not have an embedding extension, we reject $(G, H, \mathcal{H})$, and otherwise we continue.

In the next step, we check whether there is a pair of graphs H_i, H_j that have an alternation around a vertex of $\mathcal{H}$. If there is an alternation, we reject the instance, and otherwise we continue. This step can be implemented in linear time, due to Lemma 4.11.

Finally, we check the existence of PC-obstructions, which by Lemma 4.12 can be done in linear time. We accept the instance if and only if we find no PC-obstruction. The correctness of this algorithm follows from Lemma 3.8. □

Next, we deal with the instances $(G, H, \mathcal{H})$ of PEP in which G is disconnected and H is arbitrary. We use Lemma 3.9 directly and show that the two conditions of the lemma can be checked in linear time. The first condition of Lemma 3.9 can be checked in linear time by Theorem 4.13. As the proof of the next theorem shows, the second condition can be tested efficiently as well.

THEOREM 4.14. PEP *can be solved in linear time.*

PROOF. Let $(G, H, \mathcal{H})$ be an instance of PEP. Let $G_1, \dots, G_t$ be the connected components of G, let H_i be the subgraph of H induced by the vertices of G_i, and let $\mathcal{H}_i$ be $\mathcal{H}$ restricted to H_i.

By Lemma 3.9, $(G, H, \mathcal{H})$ has an embedding extension if and only if each instance $(G_i, H_i, \mathcal{H}_i)$ has an embedding extension and, for $i \neq j$, no facial cycle of $\mathcal{H}_i$ separates a pair of vertices of H_j. By Theorem 4.13, we can test in linear time whether all instances $(G_i, H_i, \mathcal{H}_i)$ have an embedding extension.

It remains to test the existence of a facial cycle of $\mathcal{H}_i$ that separates vertices of H_j. For this test, we use the component-face tree $\mathcal{CF}$ of $\mathcal{H}$. Assume that $\mathcal{CF}$ is rooted at any node representing a face of $\mathcal{H}$; call this face the *root face of* $\mathcal{H}$. A face f is an *outer face of* $\mathcal{H}_j$ if at least one child of f in $\mathcal{CF}$ is a component of H_j but the parent of f does not belong to H_j (which includes the possibility that f is the root face).

We claim that a pair of vertices of H_j is separated by a facial cycle belonging to another component of H if and only if there are at least two distinct outer faces of $\mathcal{H}_j$ in $\mathcal{CF}$. To see this, assume first that $\mathcal{H}_j$ has two distinct outer faces f_1 and f_2, and let C_1 (or C_2) be a component of H_j which is a child of f_1 (or f_2, respectively). Any path from C_1 to C_2 in $\mathcal{CF}$ visits the parent of f_1 or the parent of f_2. These parents correspond to components of H not belonging to H_j, and at least one facial cycle determined by these components separates C_1 from C_2.

Conversely, if C_1 and C_2 are components of H_j separated by a facial cycle belonging to a component C_3 of H_i ($i \neq j$), then the path in $\mathcal{CF}$ that connects C_1 to C_2 visits C_3, and in such a case it is easy to see that $\mathcal{H}_j$ has at least two outer faces.

We now describe the algorithm that tests the second condition of Lemma 3.9. We assume that each connected component of H has associated its corresponding subgraph H_i in $\mathcal{CF}$. We then process the components of H one by one, and for each component C, we check whether its parent node is an outer face of the embedding $\mathcal{H}_i$ of the subgraph H_i containing C. We accept $(G, H, \mathcal{H})$ if and only if each $\mathcal{H}_i$ has one outer face. This algorithm clearly runs in linear time. □

The algorithms for PEP we presented in this section, handling simply connected and disconnected PEGS, are nonconstructive. For simplicity, we preferred to present shorter and nonconstructive versions of these algorithms. We now briefly sketch how they can be extended to constructive linear-time algorithms.

Sketch of constructive algorithms. For the reduction from disconnected to connected PEGS, designing a constructive algorithm is rather simple. Let $(G, H, \mathcal{H})$ be a PEG and let $G_1, \dots, G_t$ be the connected components of G. Assume that we already have an embedding $\mathcal{G}_i$ for each instance $(G_i, H_i, \mathcal{H}_i)$, where H_i is the subgraph of H contained in G_i and $\mathcal{H}_i$ is the restriction of $\mathcal{H}$ to H_i. Note that each H_i may consist of several connected components. To merge the embeddings $\mathcal{G}_i$ into a single embedding $\mathcal{G}$ that extends $\mathcal{H}$, we make use of the following auxiliary graph. Create a node for each face of $\mathcal{H}$ and a node for each H_i, $i = 1, \dots, t$. Connect a face f and a component H_i if f is incident to an edge of H_i. For each edge (f, H_i) of this auxiliary graph, additionally store a pointer to some edge $e_{f,H_i} \in H_i$ such that f is to the right of e_{f,H_i}. By Lemma 3.9, this auxiliary graph is a tree T if the instance admits a solution. It can be computed from the component face tree of $\mathcal{H}$ in $O(n)$ time. We use T to guide the merging of the embeddings $\mathcal{G}_i$. Namely, for each face node f, let I_f be the set of indices i so that f is adjacent to H_i. To merge the embeddings $\mathcal{G}_i$, $i \in I_f$, we find in each of the $\mathcal{G}_i$ a subface of f and merge their face boundaries. To find a subface of f in G_i, we use the additional information stored in T. Namely, T contains the edge (f, H_i), and using the pointer stored there, we find the edge e_{f,H_i} of H_i. Recall that f lies to the right of e_{f,H_i} in $\mathcal{H}$. Let f' be the face that lies to the right of e_{f,H_i} in $\mathcal{G}_i$. Clearly, f' is a subface of f, and it can be found in $O(1)$ time. Thus, the merge step for face f can be implemented to run in $O(|I_f|)$ time, and the tree structure of T implies that the total time for merging all faces is $O(n)$.

Let us now consider the reduction from connected to biconnected PEGS. Recall that for a cutvertex x of G, an x-edge is an edge of H incident to x, and an x-block is a block of G containing at least one x-edge. Observe that the procedure TEST(x) described in the proof of Lemma 4.11 does not just check whether, around each cutvertex x, the x-blocks $B_1, \dots, B_t$ of G have a parenthetical structure, but it can actually be employed to find an ordering of $B_1, \dots, B_t$ such that the blocks can be removed one by one in such a way that the x-edges of B_i form an interval when B_i is removed. Further, the check whether a trivial H-bridge is part of a PC-obstruction actually reveals a unique face of $\mathcal{H}$ into which the block containing the H-bridge has to be embedded. We use an arbitrary such H-bridge to determine the correct face into which any block that does not contain any x-edge has to be embedded. This either gives a correct embedding or one of the conditions of Lemma 4.11 or Lemma 4.12 is violated, in which case an embedding does not exist. In the following, we assume that we have a feasible instance.

Our procedure for handling simply connected graphs is as follows. We first compute the block-cutvertex tree $\mathcal{BC}$ of G and use Lemma 2.2 to ensure that all nonlocal H-bridges are trivial. Let $G_1, \dots, G_t$ be the blocks of G and let $H_1, \dots, H_t$ be the subgraphs of H induced by $G_1, \dots, G_t$, together with the embeddings $\mathcal{H}_1, \dots, \mathcal{H}_t$ induced by $\mathcal{H}$. First, we compute in linear time an embedding extension $\mathcal{G}_i$ for each instance $(G_i, H_i, \mathcal{H}_i)$. Next, we merge embeddings $\mathcal{G}_i$ into a single embedding $\mathcal{G}$ extending $\mathcal{H}$, if possible. To this end, we need to merge the rotation systems of the embeddings $\mathcal{G}_i$ at the cutvertices of G. First, we iteratively remove each leaf block of $\mathcal{BC}$ that does not contain any vertex of H, except, possibly, for the unique cutvertex of G that it contains. Clearly, the removed blocks can easily be embedded later, as they are not subject to any constraints. The remaining instance (let us denote it again by $(G, H, \mathcal{H})$) is still connected. Moreover, by the assumption that all nonlocal H-bridges are trivial, all cutvertices of G belong to H. Therefore, every block contains at least two vertices of H (namely, every block of G that is not a leaf in $\mathcal{BC}$ contains at least two cutvertices, and every block of G that is a leaf in $\mathcal{BC}$ contains at least two vertices of H, as otherwise it would have been removed).

Let x be a cutvertex of G and let $B_1, \ldots, B_t$ be the blocks incident to x. Denote by $\mathcal{B}_i$ the embedding of B_i that has been already computed and that extends the restriction of embedding $\mathcal{H}$ to the vertices of H in B_i. Assume that the ordering of the blocks incident to x is such that $B_1, \ldots, B_k$ contain an x-edge, whereas $B_{k+1}, \ldots, B_t$ do not. We embed the blocks $B_1, \ldots, B_k$ by using a modified version of the procedure TEST(x), described in the proof of Lemma 4.11. For each i in $k+1, \ldots, t$, let $e = xy$ be any edge incident to x in B_i. Since B_i contains an H-vertex distinct from x, vertex y belongs to H as well; otherwise, e would be part of a nontrivial nonlocal H-bridge. Since edge e does not belong to H, we have that x and y belong to distinct connected components of H. We use the component face tree $\mathcal{CF}$ to find in $O(1)$ time the unique face f_i of $\mathcal{H}$ that is shared by x and y. We associate B_i with f_i. Although the choice of e is arbitrary, either all edges of B_i incident to x yield the same face or at least one of them is part of a PC-obstruction, which we can rule out by first running the checking algorithm.

We now construct the cyclic ordering of all edges of G incident to x by a single traversal of $\sigma_{\mathcal{H}}(x)$, similarly to procedure TEST(x), except we alternately visit edges and faces as they occur in counterclockwise order around x in $\mathcal{H}$.

When the procedure visits a face f, it appends the edges of all blocks associated with this face in the order as they occur in the embedding of the block. More precisely, let B_i be any block associated with f, let e be any edge of B_i incident to x, and let e' be the predecessor of e in the counterclockwise ordering of x in $\mathcal{B}_i$. Then, the counterclockwise order of the edges incident to x in $\mathcal{B}_i$ forms a sequence $e, \ldots, e'$, which we append to our global ordering of the edges of G incident to x. We do this for all blocks associated with f in an arbitrary order. When the procedure encounters the first x-edge e_i of a block B_i (recall that such an edge belongs to H and is incident to x), it appends e_i to the global ordering of x and stores the last encountered x-edge of B_i as e_i. Whenever it encounters another x-edge e' of B_i, it appends all edges between e_i and e', excluding e_i and including e', to the global ordering of the edges incident to x, then updates the last encountered x-edge of B_i to e'. When the procedure encounters the last x-edge of a block, it also inserts all remaining edges of $\mathcal{B}_i$ between the last encountered x-edge and the first x-edge of B_i. As the x-edges occur in the embedding $\mathcal{B}_i$ of each block B_i in the same order as in $\sigma_{\mathcal{H}}(x)$, each edge is inserted exactly once into the cyclic ordering. Considering the output sequence as a cyclic sequence, we have found a cyclic ordering of all edges incident to x in G. Clearly, the running time of the procedure is proportional to the number of edges incident to x in G. In addition, the ordering is such that its restriction to H yields $\sigma_{\mathcal{H}}(x)$, no two blocks alternate, and the ordering of the edges of each incident block B_i are compatible with $\mathcal{B}_i$. Finally, also the blocks that do not have an x-edge are embedded into the correct face since this face exists and thus is uniquely determined. The previously removed blocks containing at most one vertex of H—the cutvertex x—can be embedded into arbitrary faces incident to their cutvertices in reverse order of removal. Clearly, the total running time of this procedure is linear. The following theorem summarizes our results.

THEOREM 4.15. *Let* $(G, H, \mathcal{H})$ *be a* PEG. *There is a linear-time algorithm that either finds an embedding extension* $\mathcal{G}$ *of* $\mathcal{H}$ *or concludes that such an embedding does not exist.*

5. APPLICATIONS AND EXTENSIONS

In this section, we discuss several extensions of the problem PARTIALLY EMBEDDED PLANARITY. Additionally, we show that PEP has some connections to the problem of finding a *simultaneous embedding with fixed edges* of a pair of graphs. In particular, the results of this work can be used to solve this problem for a restricted class of inputs.

Problem extensions. Several generalizations of the PARTIALLY EMBEDDED PLANARITY problem naturally arise. In all of the following generalizations (denoted by G1 through G4), the input is still a PEG $(G, H, \mathcal{H})$. For the first two generalizations, we readily conclude that they are NP-complete since they contain as special cases CROSSING NUMBER and MAXIMUM PLANAR SUBGRAPH, respectively: (G1) deciding if $\mathcal{H}$ can be extended to a drawing of G with at most k crossings and (G2) deciding if at least k edges of $E(G)\backslash E(H)$ can be added to $\mathcal{H}$ while preserving planarity.

The following two additional problems generalize PEP in different directions: (G3) deciding whether G has a planar embedding $\mathcal{G}$ in which at least k edges of H are embedded as in $\mathcal{H}$ and (G4) deciding whether there is a set $F \subseteq E(H)$ of at most k edges such that $(G\backslash F, H\backslash F, \mathcal{H}\backslash F)$ is a planar PEG. We show that problems G3 and G4, called MINIMUM REROUTING PARTIALLY EMBEDDED PLANARITY and MAXIMUM PRESERVED PARTIALLY EMBEDDED PLANARITY, respectively, are NP-hard.

THEOREM 5.1. MINIMUM REROUTING PARTIALLY EMBEDDED PLANARITY *and* MAXIMUM PRESERVED PARTIALLY EMBEDDED PLANARITY *are NP-hard.*

PROOF. The proof is by reduction from STEINERTREE in planar graphs, which is known to be NP-hard [Garey and Johnson 1977]. The problem STEINERTREE in planar graphs takes as an input a planar graph $G = (V, E)$, a set $T \subset V$ of *terminals*, and an integer k and asks whether a tree $T^* = (V^*, E^*)$ exists such that (1) $V^* \subseteq V$, (2) $E^* \subseteq E$, (3) $T \subseteq V^*$, and (4) $|E^*| \leq k$.

We show how to construct an equivalent instance $(G', H, \mathcal{H}, k')$ of MAXIMUM PRESERVED PARTIALLY EMBEDDED PLANARITY, given an instance $(G = (V, E), T, k)$ of STEINERTREE in planar graphs. First, choose an embedding Γ of G and let H be the dual of Γ, with embedding $\mathcal{H}$. For each terminal $t \in T$, we add a new vertex v_t to H and prescribe it inside the face that is dual to t. This completes the construction of H and $\mathcal{H}$. Graph G' has the same vertex set as H, and its edge set is $E(H) \cup S$, where S is the edge set of any connected planar graph G_S spanning the vertices v_t. Finally, we set $k' = k$.

Now consider the problem of finding a set F of k edges of H such that $(G'\backslash F, H\backslash F, \mathcal{H}\backslash F)$ is a planar PEG. Clearly, G_S can be drawn in a planar way if and only if we choose F in such a way that all vertices v_t lie in the same face of $\mathcal{H}\backslash F$. This is equivalent to the property that the set $F^\star$ of edges dual to F is a Steiner tree in G with terminal set T. Hence, $(G', H, \mathcal{H}, k')$ is a positive instance of MAXIMUM PRESERVED PARTIALLY EMBEDDED PLANARITY if and only if (G, T, k) is a positive instance of STEINERTREE. This shows that MAXIMUM PRESERVED PARTIALLY EMBEDDED PLANARITY is NP-hard.

The reduction from an instance (G, T, k) of STEINERTREE in planar graphs to an instance $(G', H, \mathcal{H}, k')$ of MINIMUM REROUTING PARTIALLY EMBEDDED PLANARITY is analogous to the one for MAXIMUM PRESERVED PARTIALLY EMBEDDED PLANARITY. In particular, G', H, and $\mathcal{H}$ are constructed in exactly the same way; however, in such a case, we have

$k' = |E| - k$. Then, it is sufficient to observe that (i) graph G_S can be drawn in a planar way if and only if a set F of edges can be deleted in such a way that all vertices v_t lie in the same face of $\mathcal{H} \setminus F$; (ii) the set $F^\star$ of edges dual to F is a Steiner tree in G with terminal set T; and (iii) since G_S is a connected component of G', the edges of F can be reinserted without crossings into the drawing—that is, to have all vertices v_t lie in the same face of $\mathcal{H}$, it is sufficient to reroute (instead of delete) the edges in F. This shows that Minimum Rerouting Partially Embedded Planarity is NP-hard. □

In the case of Maximum Preserved Partially Embedded Planarity, we can even make H connected as follows. We connect each vertex v_t to an arbitrary vertex of its prescribed face, and we let G_S be a star graph on the vertices v_t. Thus, Maximum Preserved Partially Embedded Planarity is NP-hard even if the prescribed graph H is connected. However, this strategy does not work for Minimum Rerouting Partially Embedded Planarity, as the reduction for this problem relies on the property that every edge of each face can be removed and reinserted after drawing G_S. This is not the case if H is connected. We leave open the question whether Minimum Rerouting Partially Embedded Planarity is NP-hard if the graph H with prescribed embedding is connected.

Application to simultaneous embedding with fixed edges. The results presented in this work can be used to solve special cases of the problem simultaneous embedding with fixed edges. A *simultaneous embedding with fixed edges* (in the following called Sefe, for short) of a pair $G_1 = (V, E_1)$, $G_2 = (V, E_2)$ of graphs on the same vertex set is a pair (Γ_1, Γ_2) of drawings such that (i) Γ_i is a planar drawing of G_i, for each $i = 1, 2$; (ii) each vertex $v \in V$ is drawn on the same point in Γ_1 and in Γ_2; and (iii) each edge $(u, v) \in E_1 \cap E_2$ is represented by the same curve in Γ_1 and in Γ_2. The problem can also be generalized to simultaneous embedding of more than two graphs.

The Sefe problem is a well-studied problem in graph drawing. A lot of research has been devoted to find pairs of graph classes that always admit a Sefe and to determine how many bends are necessary for constructing a Sefe of pairs of graphs that admit one (e.g., see Angelini et al. [2012b], Di Giacomo and Liotta [2007], Erten and Kobourov [2005], Fowler et al. [2011], and Frati [2006]). Additionally, a lot of work is concerned with the algorithmic aspects of the Sefe problem. In particular, it is known that Sefe is NP-hard for two geometric graphs, where edges are restricted to be straight-line segments [Estrella-Balderrama et al. 2007] and that Sefe is NP-hard for three (or more) graphs [Gassner et al. 2006]. Polynomial-time algorithms have been designed for deciding the existence of a Sefe of two graphs, if some further assumptions are made on the input, such as if the intersection of the two input graphs is biconnected [Angelini et al. 2012a; Haeupler et al. 2013], if the input graphs are biconnected and the common graph is connected [Bläsius and Rutter 2013b], and if the connected components of the common graph are biconnected or have low degree [Bläsius et al. 2013a; Bläsius and Rutter 2013a; Schaefer 2013]. Refer to Bläsius et al. [2013b] for a comprehensive survey on this topic. Despite a large amount of research, the complexity status of the Sefe problem for two graphs remains open.

The results presented in this work allow us to solve in linear time an interesting case of the Sefe problem. Namely, Jünger and Schulz [2009] showed that two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ admit a Sefe if and only if they admit planar embeddings $\mathcal{E}_1$ and $\mathcal{E}_2$, respectively, that coincide on the intersection graph. This result, together with the results we presented on the Pep problem, implies the following theorem.

Theorem 5.2. *Let G_1 and G_2 be two graphs with the same n vertices, and let $\mathcal{G}_{1\cap 2}$ be a planar embedding of their intersection graph $G_{1\cap 2} := G_1 \cap G_2$. There exists a linear-time algorithm to decide whether G_1 and G_2 admit a* Sefe *in which the embedding of $G_{1\cap 2}$ coincides with $\mathcal{G}_{1\cap 2}$.*

PROOF. By Jünger and Schulz [2009], G_1 and G_2 admit a SEFE in which the embedding of $G_{1\cap 2}$ coincides with $\mathcal{G}_{1\cap 2}$ if and only if each of G_1 and G_2 admits a planar embedding that coincides with $\mathcal{G}_{1\cap 2}$ when restricted to the vertices and edges of $G_{1\cap 2}$. In other words, G_1 and G_2 admit a SEFE in which the embedding of $G_{1\cap 2}$ coincides with $\mathcal{G}_{1\cap 2}$ if and only if $(G_1, G_{1\cap 2}, \mathcal{G}_{1\cap 2})$ and $(G_2, G_{1\cap 2}, \mathcal{G}_{1\cap 2})$ are both YES-instances of PEP. □

Theorem 5.2 implies that deciding whether two graphs have a SEFE is a linear-time solvable problem if one of the graphs has a fixed embedding, such as if one of the two graphs is triconnected.

6. CONCLUDING REMARKS

In this article, we showed that PARTIALLY EMBEDDED PLANARITY (PEP) is a linear-time solvable problem. Problem PEP asks whether a partially embedded graph (PEG) is planar—that is, whether a planar drawing $\mathcal{H}$ of a subgraph H of a planar graph G can be extended to a planar drawing of G. To derive our linear-time algorithm, we first presented a combinatorial characterization of planar PEGS in terms of conditions on the structure of the triconnected, biconnected, and connected components of the input graph. This characterization immediately implies a polynomial-time algorithm for testing the planarity of a given PEG. The second part of the article was devoted to a careful implementation of the algorithm following from the characterization, resulting in an algorithm for PEP with optimal linear running time. Although edge compatibility exhibits a very local behavior and hence is not too difficult to enforce in linear time, numerous steps are necessary to handle cycle compatibility in linear time as well. In addition, we showed that our testing algorithm can be made constructive—that is, it can be implemented so that it finds an embedding extension for the input PEG, if one exists. Altogether, from a purely algorithmic point of view, this completely settles the problem PEP.

Further, we considered several generalizations of PEP and proved that they are NP-hard. Additionally, we showed that PEP exhibits strong connections with another well-known graph drawing problem: the SEFE problem. The results in this work immediately imply a linear-time algorithm for solving SEFE when the embedding of the intersection graph is fixed, which holds, for example, if one of the input graphs is triconnected.

In a subsequent paper, Jelínek et al. [2013] characterize the planar PEGS via forbidden substructures in the spirit of Kuratowski's theorem. In conjunction with the results presented in this work, this gives an efficient algorithm that, for a given PEG, either finds a planar extension or decides that the PEG is nonplanar and extracts an obstruction.

Open problems. The problem PEP asks for determining the extendability of planar combinatorial embeddings or, equivalently, of topological drawings of planar graphs. An obvious research direction is to consider the complexity of the extendability question for other drawing styles. It is known that completing partial straight-line drawings is NP-hard [Patrignani 2006], and it seems that the NP-hardness proof generalizes easily to poly-line drawings that admit a fixed number of bends per edge. Subsequent to the conference version of this article [Angelini et al. 2010], the problem of extending partial representations has been considered for function graphs and permutation graphs [Klavík et al. 2012a], for subclasses of chordal graphs [Klavík et al. 2012b], for interval graphs [Klavík et al. 2011], and for proper and unit interval graphs [Klavík et al. 2012c]. However, it might be interesting to consider the problems of extending, for example, orthogonal drawings [Tamassia 1987] or Manhattan-geodesic drawings [Katz et al. 2010].

A different direction for generalizing PEP would be to relax the strict condition to have a fixed embedding $\mathcal{H}$ for a subgraph H of the entire planar graph G. Gutwenger et al. [2008] consider the problem of testing the planarity of a graph with the further constraint that every vertex has an associated PQ-tree representing the possible rotations that are allowed for that vertex. The common generalization of PEP and this problem assumes that only a subgraph is constrained by such PQ-trees and the remaining edges can be inserted arbitrarily. Is it possible to decide planarity of a partially PQ-constrained graph G in polynomial time? A positive answer to the previous question has been provided in Bläsius and Rutter [2013b] for the case in which G is biconnected.

ACKNOWLEDGMENTS

This work began at the BICI Workshop on Graph Drawing, held in Bertinoro, Italy, in March 2009 and was carried out while the authors were at the Department of Applied Mathematics, Charles University, Prague.

REFERENCES

P. Angelini, G. Di Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter. 2010. Testing planarity of partially embedded graphs. In *Proceedings of SODA'10*. 202–221.

P. Angelini, G. Di Battista, F. Frati, M. Patrignani, and I. Rutter. 2012a. Testing the simultaneous embeddability of two graphs whose intersection is a biconnected or a connected graph. *Journal of Discrete Algorithms* 14, 150–172.

P. Angelini, M. Geyer, M. Kaufmann, and D. Neuwirth. 2012b. On a tree and a path with no geometric simultaneous embedding. *Journal of Graph Algorithms and Algorithms* 16, 1, 37–83.

P. Bertolazzi, G. Di Battista, and W. Didimo. 2000. Computing orthogonal drawings with the minimum number of bends. *IEEE Transactions on Computers* 49, 8, 826–840.

T. Bläsius, A. Karrer, and I. Rutter. 2013a. Simultaneous embedding: Edge orderings, relative positions, cutvertices. In *Graph Drawing*. Lecture Notes in Computer Science, Vol. 8242. Springer, 220–231.

T. Bläsius, S. G. Kobourov, and I. Rutter. 2013b. Simultaneous embedding of planar graphs. In *Handbook of Graph Drawing and Visualization*, R. Tamassia (Ed.). CRC Press.

T. Bläsius and I. Rutter. 2013a. Disconnectivity and relative positions in simultaneous embeddings. In *Graph Drawing*. Lecture Notes in Computer Science, Vol. 7704. Springer, 31–42.

T. Bläsius and I. Rutter. 2013b. Simultaneous pq-ordering with applications to constrained embedding problems. In *Proceedings of SODA'13*. 1030–1043.

J. M. Boyer and W. J. Myrvold. 2004. On the cutting edge: Simplified $O(n)$ planarity by edge addition. *Journal of Graph Algorithms and Applications* 8, 3, 241–273.

H. de Fraysseix, P. O. de Mendez, and P. Rosenstiehl. 2006. Trémaux trees and planarity. *International Journal on Foundations of Computer Science* 17, 1017–1030.

G. Demoucron, Y. Malgrange, and R. Pertuiset. 1964. Reconnaissance et construction de représentations planaires topologiques. *Rev. Francaise Recherche Opérationelle* 8, 33–34.

G. Di Battista and R. Tamassia. 1996. On-line planarity testing. *SIAM Journal on Computing* 25, 956–997.

E. Di Giacomo and G. Liotta. 2007. Simultaneous embedding of outerplanar graphs, paths, and cycles. *International Journal of Computational Geometry and Applications* 17, 2, 139–160.

C. Dornheim. 2002. Planar graphs with topological constraints. *Journal of Graph Algorithms and Applications* 6, 1, 27–66.

C. Erten and S. G. Kobourov. 2005. Simultaneous embedding of planar graphs with few bends. *Journal of Graph Algorithms and Applications* 9, 3, 347–364.

A. Estrella-Balderrama, E. Gassner, M. Jünger, M. Percan, M. Schaefer, and M. Schulz. 2007. Simultaneous geometric graph embeddings. In *Graph Drawing*. Lecture Notes in Computer Science, Vol. 4875. Springer, 280–290.

J. Fiala. 2003. NP-completeness of the edge precoloring extension problem on bipartite graphs. *Journal of Graph Theory* 43, 2, 156–160.

J. Fowler, M. Jünger, S. G. Kobourov, and M. Schulz. 2011. Characterizations of restricted pairs of planar graphs allowing simultaneous embedding with fixed edges. *Computational Geometry* 44, 8, 385–398.

F. Frati. 2006. Embedding graphs simultaneously with fixed edges. In *Graph Drawing*. Lecture Notes in Computer Science, Vol. 4372. Springer, 108–113.

M. Garey and D. Johnson. 1977. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics* 32, 4, 826–834.

E. Gassner, M. Jünger, M. Percan, M. Schaefer, and M. Schulz. 2006. Simultaneous graph embeddings with fixed edges. In *Graph-Theoretic Concepts in Computer Science*. Lecture Notes in Computer Science, Vol. 4271. Springer, 325–335.

M. Grötschel, L. Lovász, and A. Schrijver. 1988. Stable sets in graphs. In *Geometric Algorithms and Combinatorial Optimization*. Springer, 273–303.

C. Gutwenger, K. Klein, and P. Mutzel. 2008. Planarity testing and optimal edge insertion with embedding constraints. *Journal of Graph Algorithms and Applications* 12, 1, 73–95.

C. Gutwenger and P. Mutzel. 2000. A linear time implementation of SPQR-trees. In *Graph Drawing*. Lecture Notes in Computer Science, Vol. 1984. Springer, 77–90.

B. Haeupler, K. R. Jampani, and A. Lubiw. 2013. Testing simultaneous planarity when the common graph is 2-connected. *Journal of Graph Algorithms and Applications* 17, 3, 147–171.

J. Hopcroft and R. E. Tarjan. 1974. Efficient planarity testing. *Journal of the ACM* 21, 4, 549–568.

V. Jelínek, J. Kratochvíl, and I. Rutter. 2013. A Kuratowski-type theorem for planarity of partially embedded graphs. *Computational Geometry: Theory and Applications* 46, 4, 466–492. DOI:http://dx.doi.org/10.1016/j.comgeo.2012.07.005

M. Jünger and M. Schulz. 2009. Intersection graphs in simultaneous embedding with fixed edges. *Journal of Graph Algorithms and Applications* 13, 2, 205–218.

M. Juvan and B. Mohar. 2005. 2-restricted extensions of partial embeddings of graphs. *European Journal of Combinatorics* 26, 3–4, 339–375.

B. Katz, M. Krug, I. Rutter, and A. Wolff. 2010. Manhattan-geodesic embedding of planar graphs. In *Graph Drawing*. Lecture Notes in Computer Science, Vol. 5849. Springer, 207–218.

P. Klavík, J. Kratochvíl, T. Krawczyk, and B. Walczak. 2012a. Extending partial representations of function graphs and permutation graphs. In *Algorithms—ESA 2012*. Lecture Notes in Computer Science, Vol. 7501. Springer, 671–682.

P. Klavík, J. Kratochvíl, Y. Otachi, I. Rutter, T. Saitoh, M. Saumell, and T. Vyskočil. 2012c. Extending partial representations of proper and unit interval graphs. *CoRR* abs/1207.6960.

P. Klavík, J. Kratochvíl, Y. Otachi, and T. Saitoh. 2012b. Extending partial representations of subclasses of chordal graphs. In *Algorithms and Computation*. Lecture Notes in Computer Science, Vol. 7676. Springer, 444–454.

P. Klavík, J. Kratochvíl, and T. Vyskočil. 2011. Extending partial representations of interval graphs. In *Theory and Applications of Models of Computation*. Lecture Notes in Computer Science, Vol. 6648. Springer, 276–285.

L. Kowalik and M. Kurowski. 2003. Short path queries in planar graphs in constant time. In *Proceedings of STOC'03*. 143–148.

J. Kratochvîl and A. Sebo. 1997. Coloring precolored perfect graphs. *Journal of Graph Theory* 25, 207–215.

K. Kuratowski. 1930. Sur le problème des courbes gauches en topologie. *Fundamenta Mathematicae* 15, 271–283.

B. Mohar. 1999. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM Journal on Discrete Mathematics* 12, 1, 6–26.

P. Mutzel. 2003. The SPQR-tree data structure in graph drawing. In *Automata, Languages and Programming*. Lecture Notes in Computer Science, Vol. 2719. Springer, 35–46.

M. Patrignani. 2006. On extending a partial straight-line drawing. *International Journal of Foundations of Computer Science* 17, 5, 1061–1069.

J. A. La Poutré. 1994. Alpha-algorithms for incremental planarity testing. In *Proceedings of STOC'94*. 706–715.

M. Schaefer. 2013. Toward a theory of planarity: Hanani-tutte and planarity variants. *Journal of Graph Algorithms and Applications* 17, 4, 367–440.

R. Tamassia. 1987. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing* 16, 3, 421–444.

R. Tamassia. 1996. On-line planar graph embedding. *Journal of Algorithms* 21, 2, 201–239.

R. Tamassia. 1998. Constraints in graph drawing algorithms. *Constraints* 3, 1, 87–120.

R. Tamassia, G. Di Battista, and C. Batini. 1988. Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man, and Cybernetics* 18, 1, 61–79.

R. E. Tarjan. 1972. Depth first search and linear graph algorithms. *SIAM Journal on Computing* 2, 146–160.

J. Westbrook. 1992. Fast incremental planarity testing. In *Automata, Languages and Programming*. Lecture Notes in Computer Science, Vol. 623. Springer, 342–353.

Received March 2013; revised April 2014; accepted April 2014

Appendix I

On the Beer Index of Convexity and its Variants

On the Beer Index of Convexity and Its Variants*

Martin Balko[1], Vít Jelínek[2], Pavel Valtr[1], and Bartosz Walczak[3,4]

1 **Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic**
balko@kam.mff.cuni.cz, valtr@kam.mff.cuni.cz

2 **Institute for Theoretical Computer Science, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic**
jelinek@iuuk.mff.cuni.cz

3 **Theoretical Computer Science Department, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland**
walczak@tcs.uj.edu.pl

4 **School of Mathematics, Georgia Institute of Technology, Atlanta, GA, USA**

Abstract

Let S be a subset of $\mathbb{R}^d$ with finite positive Lebesgue measure. The *Beer index of convexity* $\mathrm{b}(S)$ of S is the probability that two points of S chosen uniformly independently at random see each other in S. The *convexity ratio* $\mathrm{c}(S)$ of S is the Lebesgue measure of the largest convex subset of S divided by the Lebesgue measure of S. We investigate the relationship between these two natural measures of convexity of S.

We show that every set $S \subseteq \mathbb{R}^2$ with simply connected components satisfies $\mathrm{b}(S) \leqslant \alpha\,\mathrm{c}(S)$ for an absolute constant α, provided $\mathrm{b}(S)$ is defined. This implies an affirmative answer to the conjecture of Cabello et al. asserting that this estimate holds for simple polygons.

We also consider higher-order generalizations of $\mathrm{b}(S)$. For $1 \leqslant k \leqslant d$, the *$k$-index of convexity* $\mathrm{b}_k(S)$ of $S \subseteq \mathbb{R}^d$ is the probability that the convex hull of a $(k+1)$-tuple of points chosen uniformly independently at random from S is contained in S. We show that for every $d \geqslant 2$ there is a constant $\beta(d) > 0$ such that every set $S \subseteq \mathbb{R}^d$ satisfies $\mathrm{b}_d(S) \leqslant \beta\,\mathrm{c}(S)$, provided $\mathrm{b}_d(S)$ exists. We provide an almost matching lower bound by showing that there is a constant $\gamma(d) > 0$ such that for every $\varepsilon \in (0, 1]$ there is a set $S \subseteq \mathbb{R}^d$ of Lebesgue measure one satisfying $\mathrm{c}(S) \leqslant \varepsilon$ and $\mathrm{b}_d(S) \geqslant \gamma \frac{\varepsilon}{\log_2 1/\varepsilon} \geqslant \gamma \frac{\mathrm{c}(S)}{\log_2 1/\mathrm{c}(S)}$.

1998 ACM Subject Classification F.2.2 Geometrical problems and computations

Keywords and phrases Beer index of convexity, convexity ratio, convexity measure, visibility

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.406

1 Introduction

For positive integers k and d and a Lebesgue measurable set $S \subseteq \mathbb{R}^d$, we use $\lambda_k(S)$ to denote the k-dimensional Lebesgue measure of S. We omit the subscript k when it is clear from the context. We also write 'measure' instead of 'Lebesgue measure', as we do not use any other measure in the paper.

For a set $S \subseteq \mathbb{R}^d$, let $\mathrm{smc}(S)$ denote the supremum of the measures of convex subsets of S. Since all convex subsets of $\mathbb{R}^d$ are measurable [12], the value of $\mathrm{smc}(S)$ is well defined.

* The first three authors were supported by the grant GAČR 14-14179S. The first author acknowledges the support of the Grant Agency of the Charles University, GAUK 690214 and the project SVV-2014-260103 (Discrete Models and Algorithms). The last author was supported by the Ministry of Science and Higher Education of Poland *Mobility Plus* grant 911/MOB/2012/0.

31st International Symposium on Computational Geometry (SoCG'15).
Editors: Lars Arge and János Pach; pp. 406–420
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Moreover, Goodman's result [9] implies that the supremum is achieved on compact sets S, hence it can be replaced by maximum in this case. When S has finite positive measure, let $\mathrm{c}(S)$ be defined as $\mathrm{smc}(S)/\lambda_d(S)$. We call the parameter $\mathrm{c}(S)$ the *convexity ratio* of S.

For two points $A, B \in \mathbb{R}^d$, let $\overline{AB}$ denote the closed line segment with endpoints A and B. Let S be a subset of $\mathbb{R}^d$. We say that points $A, B \in S$ are *visible* one from the other or *see* each other in S if the line segment $\overline{AB}$ is contained in S. For a point $A \in S$, we use $\mathrm{Vis}(A, S)$ to denote the set of points that are visible from A in S. More generally, for a subset T of S, we use $\mathrm{Vis}(T, S)$ to denote the set of points that are visible in S from T. That is, $\mathrm{Vis}(T, S)$ is the set of points $A \in S$ for which there is a point $B \in T$ such that $\overline{AB} \subseteq S$.

Let $\mathrm{Seg}(S)$ denote the set $\{(A, B) \in S \times S \colon \overline{AB} \subseteq S\} \subseteq (\mathbb{R}^d)^2$, which we call the *segment set* of S. For a set $S \subseteq \mathbb{R}^d$ with finite positive measure and with measurable $\mathrm{Seg}(S)$, we define the parameter $\mathrm{b}(S) \in [0, 1]$ by

$$\mathrm{b}(S) := \frac{\lambda_{2d}(\mathrm{Seg}(S))}{\lambda_d(S)^2}.$$

If S is not measurable, or if its measure is not positive and finite, or if $\mathrm{Seg}(S)$ is not measurable, we leave $\mathrm{b}(S)$ undefined. Note that if $\mathrm{b}(S)$ is defined for a set S, then $\mathrm{c}(S)$ is defined as well.

We call $\mathrm{b}(S)$ the *Beer index of convexity* (or just *Beer index*) of S. It can be interpreted as the probability that two points A and B of S chosen uniformly independently at random see each other in S.

1.1 Previous results

The Beer index was introduced in the 1970s by Beer [2, 3, 4], who called it 'the index of convexity'. Beer was motivated by studying the continuity properties of $\lambda(\mathrm{Vis}(A, S))$ as a function of A. For polygonal regions, an equivalent parameter was later independently defined by Stern [19], who called it 'the degree of convexity'. Stern was motivated by the problem of finding a computationally tractable way to quantify how close a given set is to being convex. He showed that the Beer index of a polygon P can be approximated by a Monte Carlo estimation. Later, Rote [17] showed that for a polygonal region P with n edges the Beer index can be evaluated in polynomial time as a sum of $O(n^9)$ closed-form expressions.

Cabello et al. [7] have studied the relationship between the Beer index and the convexity ratio, and applied their results in the analysis of their near-linear-time approximation algorithm for finding the largest convex subset of a polygon. We describe some of their results in more detail in Subsection 1.3.

1.2 Terminology and notation

We assume familiarity with basic topological notions such as path-connectedness, simple connectedness, Jordan curve, etc. The reader can find these definitions, for example, in Prasolov's book [16].

Let ∂S, S°, and $\overline{S}$ denote the boundary, the interior, and the closure of a set S, respectively. For a point $A \in \mathbb{R}^2$ and $\varepsilon > 0$, let $\mathcal{N}_\varepsilon(A)$ denote the open disc centered at A with radius ε. For a set $X \subseteq \mathbb{R}^2$ and $\varepsilon > 0$, let $\mathcal{N}_\varepsilon(X) = \bigcup_{A \in X} \mathcal{N}_\varepsilon(A)$. A *neighborhood* of a point $A \in \mathbb{R}^2$ or a set $X \subseteq \mathbb{R}^2$ is a set of the form $\mathcal{N}_\varepsilon(A)$ or $\mathcal{N}_\varepsilon(X)$, respectively, for some $\varepsilon > 0$.

A closed interval with endpoints a and b is denoted by $[a, b]$. Intervals $[a, b]$ with $a > b$ are considered empty. For a point $A \in \mathbb{R}^2$, we use $x(A)$ and $y(A)$ to denote the x-coordinate and the y-coordinate of A, respectively.

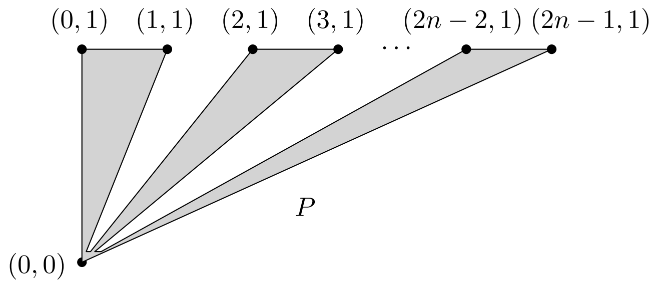


Figure 1 A star-shaped polygon P with $\mathrm{b}(P) \geqslant \frac{1}{n} - \varepsilon$ and $\mathrm{c}(P) \leqslant \frac{1}{n}$. The polygon P with $4n-1$ vertices is a union of n triangles $(0,0)(2i,1)(2i+1,1)$, $i = 0,\dots,n-1$, and of a triangle $(0,0)(0,\delta)((2n-1)\delta,\delta)$, where δ is very small.

A *polygonal curve* Γ in $\mathbb{R}^d$ is a curve specified by a sequence $(A_1,\dots,A_n)$ of points of $\mathbb{R}^d$ such that Γ consists of the line segments connecting the points A_i and A_{i+1} for $i = 1,\dots,n-1$. If $A_1 = A_n$, then the polygonal curve Γ is *closed*. A polygonal curve that is not closed is called a *polygonal line*.

A set $X \subseteq \mathbb{R}^2$ is *polygonally connected*, or *p-connected* for short, if any two points of X can be connected by a polygonal line in X, or equivalently, by a self-avoiding polygonal line in X. For a set X, the relation "A and B can be connected by a polygonal line in X" is an equivalence relation on X, and its equivalence classes are the *p-components* of X. A set S is *p-componentwise simply connected* if every p-component of S is simply connected.

A *line segment* in $\mathbb{R}^d$ is a bounded convex subset of a line. A *closed line segment* includes both endpoints, while an *open line segment* excludes both endpoints. For two points A and B in $\mathbb{R}^d$, we use AB to denote the open line segment with endpoints A and B. A closed line segment with endpoints A and B is denoted by $\overline{AB}$.

We say that a set $S \subseteq \mathbb{R}^d$ is *star-shaped* if there is a point $C \in S$ such that $\mathrm{Vis}(C, S) = S$. That is, a star-shaped set S contains a point which sees the entire S. Similarly, we say that a set S is *weakly star-shaped* if S contains a line segment ℓ such that $\mathrm{Vis}(\ell, S) = S$.

1.3 Results

We start with a few simple observations. Let S be a subset of $\mathbb{R}^2$ such that $\mathrm{Seg}(S)$ is measurable. For every $\varepsilon > 0$, S contains a convex subset K of measure at least $(\mathrm{c}(S)-\varepsilon)\lambda_2(S)$. Two random points of S both belong to K with probability at least $(\mathrm{c}(S) - \varepsilon)^2$, hence $\mathrm{b}(S) \geqslant (\mathrm{c}(S) - \varepsilon)^2$. This yields $\mathrm{b}(S) \geqslant \mathrm{c}(S)^2$. This simple lower bound on $\mathrm{b}(S)$ is tight, as shown by a set S which is a disjoint union of a single large convex component and a large number of small components of negligible size.

It is more challenging to find an upper bound on $\mathrm{b}(S)$ in terms of $\mathrm{c}(S)$, possibly under additional assumptions on the set S. This is the general problem addressed in this paper.

As a motivating example, observe that a set S consisting of n disjoint convex components of the same size satisfies $\mathrm{b}(S) = \mathrm{c}(S) = \frac{1}{n}$. It is easy to modify this example to obtain, for any $\varepsilon > 0$, a simple star-shaped polygon P with $\mathrm{b}(P) \geqslant \frac{1}{n} - \varepsilon$ and $\mathrm{c}(P) \leqslant \frac{1}{n}$, see Figure 1. This shows that $\mathrm{b}(S)$ cannot be bounded from above by a sublinear function of $\mathrm{c}(S)$, even for simple polygons S.

For weakly star-shaped polygons, Cabello et al. [7] showed that the above example is essentially optimal, providing the following linear upper bound on $\mathrm{b}(S)$.

▶ **Theorem 1** ([7, Theorem 5]). *For every weakly star-shaped simple polygon P, we have $\mathrm{b}(P) \leqslant 18\,\mathrm{c}(P)$.*

For polygons that are not weakly star-shaped, Cabello et al. [7] gave a superlinear bound.

▶ **Theorem 2** ([7, Theorem 6]). *Every simple polygon P satisfies*

$$\mathrm{b}(P) \leqslant 12\,\mathrm{c}(P)\left(1+\log_2\frac{1}{\mathrm{c}(P)}\right).$$

Moreover, Cabello et al. [7] conjectured that even for a general simple polygon P, $\mathrm{b}(P)$ can be bounded from above by a linear function of $\mathrm{c}(P)$. The next theorem, which is the first main result of this paper, confirms this conjecture. Recall that $\mathrm{b}(S)$ is defined for a set S if and only if S has finite positive measure and $\mathrm{Seg}(S)$ is measurable. Recall also that a set is p-componentwise simply connected if its polygonally-connected components are simply connected. In particular, every simply connected set is p-componentwise simply connected.

▶ **Theorem 3.** *Every p-componentwise simply connected set $S \subseteq \mathbb{R}^2$ whose $\mathrm{b}(S)$ is defined satisfies $\mathrm{b}(S) \leqslant 180\,\mathrm{c}(S)$.*

It is clear that every simple polygon satisfies the assumptions of Theorem 3, hence we directly obtain the following, which confirms the conjecture of Cabello et al. [7].

▶ **Corollary 4.** *Every simple polygon $P \subseteq \mathbb{R}^2$ satisfies $\mathrm{b}(P) \leqslant 180\,\mathrm{c}(P)$.*

The main restriction in Theorem 3 is the assumption that S is p-componentwise simply connected. This assumption cannot be omitted, as shown by the set $S = [0,1]^2 \smallsetminus \mathbb{Q}^2$, where it is easy to verify that $\mathrm{c}(S) = 0$ and $\mathrm{b}(S) = 1$.

A related construction shows that Theorem 3 fails in higher dimensions. To see this, consider again the set $S = [0,1]^2 \smallsetminus \mathbb{Q}^2$, and define a set $S' \subseteq \mathbb{R}^3$ by

$$S' := \{(tx, ty, t) \colon t \in [0,1] \text{ and } (x,y) \in S\}.$$

Again, it is easy to verify that $\mathrm{c}(S') = 0$ and $\mathrm{b}(S') = 1$, although S' is simply connected, even star-shaped.

Despite these examples, we will show that meaningful analogues of Theorem 3 for higher dimensions and for sets that are not p-componentwise simply connected are possible. The key is to use higher-order generalizations of the Beer index, which we introduce now.

For a set $S \subseteq \mathbb{R}^d$, we define the set $\mathrm{Simp}_k(S) \subseteq (\mathbb{R}^d)^{k+1}$ by

$$\mathrm{Simp}_k(S) := \{(A_0, \ldots, A_k) \in S^{k+1} \colon \mathrm{Conv}(\{A_0, \ldots, A_k\}) \subseteq S\},$$

where the operator Conv denotes the convex hull of a set of points. We call $\mathrm{Simp}_k(S)$ the *k-simplex set* of S. Note that $\mathrm{Simp}_1(S) = \mathrm{Seg}(S)$.

For an integer $k \in \{1, 2, \ldots, d\}$ and a set $S \subseteq \mathbb{R}^d$ with finite positive measure and with measurable $\mathrm{Simp}_k(S)$, we define $\mathrm{b}_k(S)$ by

$$\mathrm{b}_k(S) := \frac{\lambda_{(k+1)d}(\mathrm{Simp}_k(S))}{\lambda_d(S)^{k+1}}.$$

Note that $\mathrm{b}_1(S) = \mathrm{b}(S)$. We call $\mathrm{b}_k(S)$ the *k-index of convexity* of S. We again leave $\mathrm{b}_k(S)$ undefined if S or $\mathrm{Simp}_k(S)$ is non-measurable, or if the measure of S is not finite and positive.

We can view $\mathrm{b}_k(S)$ as the probability that the convex hull of $k+1$ points chosen from S uniformly independently at random is contained in S. For any $S \subseteq \mathbb{R}^d$, we have $\mathrm{b}_1(S) \geqslant \mathrm{b}_2(S) \geqslant \cdots \geqslant \mathrm{b}_d(S)$, provided all the $\mathrm{b}_k(S)$ are defined.

We remark that the set $S = [0,1]^d \smallsetminus \mathbb{Q}^d$ satisfies $\mathrm{c}(S) = 0$ and $\mathrm{b}_1(S) = \mathrm{b}_2(S) = \cdots = \mathrm{b}_{d-1}(S) = 1$. Thus, for a general set $S \subseteq \mathbb{R}^d$, only the d-index of convexity can conceivably admit a nontrivial upper bound in terms of $\mathrm{c}(S)$. Our next result shows that such an upper bound on $\mathrm{b}_d(S)$ exists and is linear in $\mathrm{c}(S)$.

▶ **Theorem 5.** *For every $d \geqslant 2$, there is a constant $\beta = \beta(d) > 0$ such that every set $S \subseteq \mathbb{R}^d$ with defined $\mathrm{b}_d(S)$ satisfies $\mathrm{b}_d(S) \leqslant \beta\,\mathrm{c}(S)$.*

We do not know if the linear upper bound in Theorem 5 is best possible. We can, however, construct examples showing that the bound is optimal up to a logarithmic factor. This is our last main result.

▶ **Theorem 6.** *For every $d \geqslant 2$, there is a constant $\gamma = \gamma(d) > 0$ such that for every $\varepsilon \in (0,1]$, there is a set $S \subseteq \mathbb{R}^d$ satisfying $\mathrm{c}(S) \leqslant \varepsilon$ and $\mathrm{b}_d(S) \geqslant \gamma\frac{\varepsilon}{\log_2 1/\varepsilon}$, and in particular, we have $\mathrm{b}_d(S) \geqslant \gamma\frac{\mathrm{c}(S)}{\log_2 1/\,\mathrm{c}(S)}$.*

In this extended abstract, some proofs have been omitted due to space constraints. The omitted proofs can be found in the full version of this paper [1].

2 Bounding the mutual visibility in the plane

The goal of this section is to prove Theorem 3. Since the proof is rather long and complicated, let us first present a high-level overview of its main ideas.

We first show that it is sufficient to prove the estimate from Theorem 3 for bounded open simply connected sets. This is formalized by the next lemma, whose proof is omitted.

▶ **Lemma 7.** *Let $\alpha > 0$ be a constant such that every open bounded simply connected set $T \subseteq \mathbb{R}^2$ satisfies $\mathrm{b}(T) \leqslant \alpha\,\mathrm{c}(T)$. It follows that every p-componentwise simply connected set $S \subseteq \mathbb{R}^2$ with defined $\mathrm{b}(S)$ satisfies $\mathrm{b}(S) \leqslant \alpha\,\mathrm{c}(S)$.*

Suppose now that S is a bounded open simply connected set. We seek a bound of the form $\mathrm{b}(S) = O(\mathrm{c}(S))$. This is equivalent to a bound of the form $\lambda_4(\mathrm{Seg}(S)) = O(\mathrm{smc}(S)\lambda_2(S))$. We therefore need a suitable upper bound on $\lambda_4(\mathrm{Seg}(S))$.

We first choose in S a *diagonal* ℓ (i.e., an inclusion-maximal line segment in S), and show that the set $S \setminus \ell$ is a union of two open simply connected sets S_1 and S_2 (Lemma 10). It is not hard to show that the segments in S that cross the diagonal ℓ contribute to $\lambda_4(\mathrm{Seg}(S))$ by at most $O(\mathrm{smc}(S)\lambda_2(S))$ (Lemma 14). Our main task is to bound the measure of $\mathrm{Seg}(S_i \cup \ell)$ for $i = 1, 2$. The two sets $S_i \cup \ell$ are what we call *rooted sets*. Informally, a rooted set is a union of a simply connected open set S' and an open segment $r \subseteq \partial S'$, called the root.

To bound $\lambda_4(\mathrm{Seg}(R))$ for a rooted set R with root r, we partition R into *levels* $L_1, L_2, \ldots,$ where L_k contains the points of R that can be connected to r by a polygonal line with k segments, but not by a polygonal line with $k-1$ segments. Each segment in R is contained in a union $L_i \cup L_{i+1}$ for some $i \geqslant 1$. Thus, a bound of the form $\lambda_4(\mathrm{Seg}(L_i \cup L_{i+1})) = O(\mathrm{smc}(R)\lambda_2(L_i \cup L_{i+1}))$ implies the required bound for $\lambda_4(\mathrm{Seg}(R))$.

We will show that each p-component of $L_i \cup L_{i+1}$ is a rooted set, with the extra property that all its points are reachable from its root by a polygonal line with at most two segments (Lemma 11). To handle such sets, we will generalize the techniques that Cabello et al. [7] have used to handle weakly star-shaped sets in their proof of Theorem 1. We will assign to every point $A \in R$ a set $\mathfrak{T}(A)$ of measure $O(\mathrm{smc}(R))$, such that for every $(A, B) \in \mathrm{Seg}(R)$, we have either $B \in \mathfrak{T}(A)$ or $A \in \mathfrak{T}(B)$ (Lemma 13). From this, Theorem 3 will follow easily.

To proceed with the proof of Theorem 3 for bounded open simply connected sets, we need a few auxiliary lemmas.

▶ **Lemma 8.** *For every positive integer d, if S is an open subset of $\mathbb{R}^d$, then the set $\mathrm{Seg}(S)$ is open and the set $\mathrm{Vis}(A, S)$ is open for every point $A \in S$.*

Proof. Choose a pair of points $(A, B) \in \mathrm{Seg}(S)$. Since S is open and $\overline{AB}$ is compact, there is $\varepsilon > 0$ such that $\mathcal{N}_\varepsilon(\overline{AB}) \subseteq S$. Consequently, for any $A' \in \mathcal{N}_\varepsilon(A)$ and $B' \in \mathcal{N}_\varepsilon(B)$, we have $\overline{A'B'} \subseteq S$, that is, $(A', B') \in \mathrm{Seg}(S)$. This shows that the set $\mathrm{Seg}(S)$ is open. If we fix $A' = A$, then it follows that the set $\mathrm{Vis}(A, S)$ is open. ◀

▶ **Lemma 9.** *Let S be a simply connected subset of $\mathbb{R}^2$ and let ℓ and ℓ' be line segments in S. It follows that the set $\mathrm{Vis}(\ell', S) \cap \ell$ is a (possibly empty) subsegment of ℓ.*

Proof. The statement is trivially true if ℓ and ℓ' intersect or have the same supporting line, or if $\mathrm{Vis}(\ell', S) \cap \ell$ is empty. Suppose that these situations do not occur. Let $A, B \in \ell$ and $A', B' \in \ell'$ be such that $\overline{AA'}, \overline{BB'} \subseteq S$. The points A, A', B', B form a (possibly self-intersecting) tetragon Q whose boundary is contained in S. Since S is simply connected, the interior of Q is contained in S. If Q is not self-intersecting, then clearly $\overline{AB} \subseteq \mathrm{Vis}(\ell', S)$. Otherwise, $\overline{AA'}$ and $\overline{BB'}$ have a point D in common, and every point $C \in AB$ is visible in R from the point $C' \in A'B'$ such that $D \in \overline{CC'}$. This shows that $\mathrm{Vis}(\ell', S) \cap \ell$ is a convex subset and hence a subsegment of ℓ. ◀

Now, we define rooted sets and their tree-structured decomposition, and we explain how they arise in the proof of Theorem 3.

A set $S \subseteq \mathbb{R}^2$ is *half-open* if every point $A \in S$ has a neighborhood $\mathcal{N}_\varepsilon(A)$ that satisfies one of the following two conditions:

1. $\mathcal{N}_\varepsilon(A) \subseteq S$,
2. $\mathcal{N}_\varepsilon(A) \cap \partial S$ is a diameter of $\mathcal{N}_\varepsilon(A)$ splitting it into two subsets, one of which (including the diameter) is $\mathcal{N}_\varepsilon(A) \cap S$ and the other (excluding the diameter) is $\mathcal{N}_\varepsilon(A) \smallsetminus S$.

The condition 1 holds for points $A \in S^\circ$, while the condition 2 holds for points $A \in \partial S$. A set $R \subseteq \mathbb{R}^2$ is a *rooted set* if the following conditions are satisfied:

1. R is bounded,
2. R is p-connected and simply connected,
3. R is half-open,
4. $R \cap \partial R$ is an open line segment.

The open line segment $R \cap \partial R$ is called the *root* of R. Every rooted set, as the union of a non-empty open set and an open line segment, is measurable and has positive measure.

A *diagonal* of a set $S \subseteq \mathbb{R}^2$ is a line segment contained in S that is not a proper subset of any other line segment contained in S. Clearly, if S is open, then every diagonal of S is an open line segment. It is easy to see that the root of a rooted set is a diagonal. The following lemma allows us to use a diagonal to split a bounded open simply connected subset of $\mathbb{R}^2$ into two rooted sets. It is intuitively clear, and its formal proof is omitted.

▶ **Lemma 10.** *Let S be a bounded open simply connected subset of $\mathbb{R}^2$, and let ℓ be a diagonal of S. It follows that the set $S \smallsetminus \ell$ has two p-components S_1 and S_2. Moreover, $S_1 \cup \ell$ and $S_2 \cup \ell$ are rooted sets, and ℓ is their common root.*

Let R be a rooted set. For a positive integer k, the *kth level* L_k of R is the set of points of R that can be connected to the root of R by a polygonal line in R consisting of k segments but cannot be connected to the root of R by a polygonal line in R consisting of fewer than k segments. We consider a degenerate one-vertex polygonal line as consisting of one degenerate segment, so the root of R is part of L_1. Thus $L_1 = \mathrm{Vis}(r, R)$, where r denotes the root of R. A *k-body* of R is a p-component of L_k. A *body* of R is a k-body of R for some k. See Figure 2 for an example of a rooted set and its partitioning into levels and bodies.

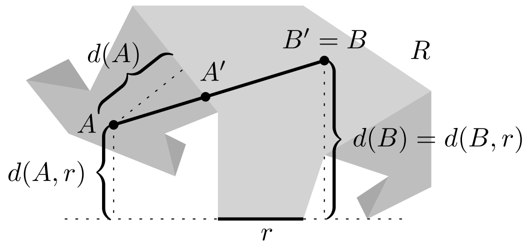


Figure 2 Example of a rooted set R partitioned into six bodies. The three levels of R are distinguished with three shades of gray. The segment $A'B'$ is the base segment of $\overline{AB}$.

We say that a rooted set P is *attached* to a set $Q \subseteq \mathbb{R}^2 \setminus P$ if the root of P is subset of the interior of $P \cup Q$. The following lemma explains the structure of levels and bodies. Although it is intuitively clear, its formal proof requires quite a lot of work and is omitted.

▶ **Lemma 11.** *Let R be a rooted set and $(L_k)_{k\geqslant 1}$ be its partition into levels. It follows that*
1. *$R = \bigcup_{k\geqslant 1} L_k$; consequently, R is the union of all its bodies;*
2. *every body P of R is a rooted set such that $P = \operatorname{Vis}(r, P)$, where r denotes the root of P;*
3. *L_1 is the unique 1-body of R, and the root of L_1 is the root of R;*
4. *every j-body P of R with $j \geqslant 2$ is attached to a unique $(j-1)$-body of R.*

Lemma 11 yields a tree structure on the bodies of R. The root of this tree is the unique 1-body L_1 of R, called the *root body* of R. For a k-body P of R with $k \geqslant 2$, the parent of P in the tree is the unique $(k-1)$-body of R that P is attached to, called the *parent body* of P.

▶ **Lemma 12.** *Let R be a rooted set, $(L_k)_{k\geqslant 1}$ be the partition of R into levels, ℓ be a closed line segment in R, and $k \geqslant 1$ be minimum such that $\ell \cap L_k \neq \emptyset$. It follows that $\ell \subseteq L_k \cup L_{k+1}$, $\ell \cap L_k$ is a subsegment of ℓ contained in a single k-body P of R, and $\ell \cap L_{k+1}$ consists of at most two subsegments of ℓ each contained in a single $(k+1)$-body whose parent body is P.*

Proof. The definition of the levels directly yields $\ell \subseteq L_k \cup L_{k+1}$. The segment ℓ splits into subsegments each contained in a single k-body or $(k+1)$-body of R. By Lemma 11, the bodies of any two consecutive of these subsegments are in the parent-child relation of the body tree. This implies that $\ell \cap L_k$ lies within a single k-body P. By Lemma 9, $\ell \cap L_k$ is a subsegment of ℓ. Consequently, $\ell \cap L_{k+1}$ consists of at most two subsegments. ◀

In the setting of Lemma 12, we call the subsegment $\ell \cap L_k$ of ℓ the *base segment* of ℓ, and we call the body P that contains $\ell \cap L_k$ the *base body* of ℓ. See Figure 2 for an example.

The following lemma is the crucial part of the proof of Theorem 3.

▶ **Lemma 13.** *If R is a rooted set, then every point $A \in R$ can be assigned a measurable set $\mathfrak{T}(A) \subseteq \mathbb{R}^2$ so that the following is satisfied:*
1. *$\lambda_2(\mathfrak{T}(A)) < 87 \operatorname{smc}(R)$;*
2. *for every line segment $\overline{BC}$ in R, we have either $B \in \mathfrak{T}(C)$ or $C \in \mathfrak{T}(B)$;*
3. *the set $\{(A, B) \colon A \in R \text{ and } B \in \mathfrak{T}(A)\}$ is measurable.*

Proof. Let P be a body of R with the root r. First, we show that P is entirely contained in one closed half-plane defined by the supporting line of r. Let h^- and h^+ be the two open half-planes defined by the supporting line of r. According to the definition of a rooted set, the sets $\{D \in r \colon \exists \varepsilon > 0 \colon \mathcal{N}_\varepsilon(D) \cap h^- = \mathcal{N}_\varepsilon(D) \cap (P \setminus r)\}$ and $\{D \in r \colon \exists \varepsilon > 0 \colon \mathcal{N}_\varepsilon(D) \cap h^+ = \mathcal{N}_\varepsilon(D) \cap (P \setminus r)\}$ are open and partition the entire r, hence one of them must be empty. This

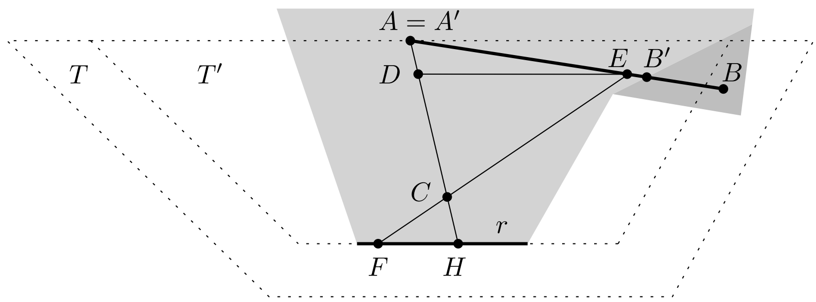


Figure 3 Illustration for the proof of Claim 1 in the proof of Lemma 13.

implies that the segments connecting r to $P \setminus r$ lie all in h^- or all in h^+. Since $P = \mathrm{Vis}(r, P)$, we conclude that $P \subseteq h^-$ or $P \subseteq h^+$.

According to the above, we can rotate and translate the set R so that r lies on the x-axis and P lies in the half-plane $\{B \in \mathbb{R}^2 \colon y(B) \geqslant 0\}$. For a point $A \in R$, we use $d(A, r)$ to denote the y-coordinate of A after such a rotation and translation of R. We use $d(A)$ to denote $d(A, r)$ where r is the root of the body of A. It follows that $d(A) \geqslant 0$ for every $A \in R$.

Let $\gamma \in (0, 1)$ be a fixed constant whose value will be specified at the end of the proof. For a point $A \in R$, we define the sets

$$\mathfrak{V}_1(A) := \{B \in \mathrm{Vis}(A, S) \colon |A'B'| \geqslant \gamma|AB|,\ A \in \mathrm{Vis}(r'', R),\ d(A', r'') \geqslant d(B', r'')\},$$
$$\mathfrak{V}_2(A) := \{B \in \mathrm{Vis}(A, S) \colon |A'B'| \geqslant \gamma|AB|,\ A \notin \mathrm{Vis}(r'', R),\ d(A', r'') \geqslant d(B', r'')\},$$
$$\mathfrak{V}_3(A) := \{B \in \mathrm{Vis}(A, S) \colon |A'B'| < \gamma|AB|,\ |AA'| \geqslant |BB'|\},$$

where r'' denotes the root of the base body of $\overline{AB}$ and A' and B' denote the endpoints of the base segment of $\overline{AB}$ such that $|AA'| < |AB'|$. These sets are pairwise disjoint, and we have $A \in \bigcup_{i=1}^{3} \mathfrak{V}_i(B)$ or $B \in \bigcup_{i=1}^{3} \mathfrak{V}_i(A)$ for every line segment $\overline{AB}$ in R. If for some $B \in \bigcup_{i=1}^{3} \mathfrak{V}_i(A)$ the point A lies on r'', then we have $B \in \mathfrak{V}_1(A)$ and $\mathfrak{V}_1(A) \subseteq r''$.

For the rest of the proof, we fix a point $A \in R$. We show that the union $\bigcup_{i=1}^{3} \mathfrak{V}_i(A)$ is contained in a measurable set $\mathfrak{T}(A) \subseteq \mathbb{R}^2$ with $\lambda_2(\mathfrak{T}(A)) < 87\,\mathrm{smc}(R)$ that is the union of three trapezoids. We let P be the body of A and r be the root of P. If P is a k-body with $k \geqslant 2$, then we use r' to denote the root of the parent body of P.

▶ Claim 1. *$\mathfrak{V}_1(A)$ is contained in a trapezoid $\mathfrak{T}_1(A)$ with area $6\gamma^{-2}\,\mathrm{smc}(R)$.*

Let H be a point of r such that $\overline{AH} \subseteq R$. Let T' be the r-parallel trapezoid of height $d(A)$ with bases of length $\frac{8\,\mathrm{smc}(R)}{d(A)}$ and $\frac{4\,\mathrm{smc}(R)}{d(A)}$ such that A is the center of the larger base and H is the center of the smaller base. The homothety with center A and ratio γ^{-1} transforms T' into the trapezoid $T := A + \gamma^{-1}(T' - A)$. Since the area of T' is $6\,\mathrm{smc}(R)$, the area of T is $6\gamma^{-2}\,\mathrm{smc}(R)$. We show that $\mathfrak{V}_1(A) \subseteq T$. See Figure 3 for an illustration.

Let B be a point in $\mathfrak{V}_1(A)$. Using similar techniques to the ones used by Cabello et al. [7] in the proof of Theorem 1, we show that $B \in T$. Let $A'B'$ be the base segment of $\overline{AB}$ such that $|AA'| < |AB'|$. Since $B \in \mathfrak{V}_1(A)$, we have $|A'B'| \geqslant \gamma|AB|$, $A \in \mathrm{Vis}(r'', R)$, and $d(B, r'') \leqslant d(A, r'')$, where r'' denotes the root of the base level of $\overline{AB}$. Since A is visible from r'' in R, the base body of $\overline{AB}$ is the body of A and thus $A = A'$ and $r = r''$. As we have observed, every point $C \in \{A\} \cup AB'$ satisfies $d(C, r) = d(C) \geqslant 0$.

Let $\varepsilon > 0$. There is a point $E \in AB'$ such that $|B'E| < \varepsilon$. Since E lies on the base segment of $\overline{AB}$, there is $F \in r$ such that $\overline{EF} \subseteq R$. It is possible to choose F so that $\overline{AH}$ and $\overline{EF}$ have a point C in common where $C \neq F, H$. Let D be a point of $\overline{AH}$ with $d(D) = d(E)$. The point D exists, as $d(H) = 0 \leqslant d(E) \leqslant d(A)$. The points A, E, F, H

form a self-intersecting tetragon Q whose boundary is contained in R. Since R is simply connected, the interior of Q is contained in R and the triangles ACE and CFH have area at most $\mathrm{smc}(R)$.

The triangle ACE is partitioned into triangles ADE and CDE with areas $\frac{1}{2}(d(A) - d(D))|DE|$ and $\frac{1}{2}(d(D) - d(C))|DE|$, respectively. Therefore, we have $\frac{1}{2}(d(A) - d(C))|DE| = \lambda_2(ACE) \leqslant \mathrm{smc}(R)$. This implies

$$|DE| \leqslant \frac{2\,\mathrm{smc}(R)}{d(A) - d(C)}.$$

For the triangle CFH, we have $\frac{1}{2}d(C)|FH| = \lambda_2(CFH) \leqslant \mathrm{smc}(R)$. By the similarity of the triangles CFH and CDE, we have $|FH| = |DE|d(C)/(d(E) - d(C))$ and therefore

$$|DE| \leqslant \frac{2\,\mathrm{smc}(R)}{d(C)^2}(d(E) - d(C)).$$

Since the first upper bound on $|DE|$ is increasing in $d(C)$ and the second is decreasing in $d(C)$, the minimum of the two is maximized when they are equal, that is, when $d(C) = d(A)d(E)/(d(A) + d(E))$. Then we obtain $|DE| \leqslant \frac{2\,\mathrm{smc}(R)}{d(A)^2}(d(A) + d(E))$. This and $0 \leqslant d(E) \leqslant d(A)$ imply $E \in T'$. Since ε can be made arbitrarily small and T' is compact, we have $B' \in T'$. Since $|AB'| \geqslant \gamma|AB|$, we conclude that $B \in T$. This completes the proof of Claim 1.

▶ Claim 2. *$\mathfrak{V}_2(A)$ is contained in a trapezoid $\mathfrak{T}_2(A)$ with area $3(1-\gamma)^{-2}\gamma^{-2}\,\mathrm{smc}(R)$.*

We assume the point A is not contained in the first level of R, as otherwise $\mathfrak{V}_2(A)$ is empty. Let p be the r'-parallel line that contains the point A and let q be the supporting line of r. Let p^+ and q^+ denote the closed half-planes defined by p and q, respectively, such that $r' \subseteq p^+$ and $A \notin q^+$. Let O be the intersection point of p and q.

Let $T' \subseteq p^+ \cap q^+$ be the trapezoid of height $d(A, r')$ with one base of length $\frac{4\,\mathrm{smc}(R)}{(1-\gamma)^2 d(A,r')}$ on p, the other base of length $\frac{2\,\mathrm{smc}(R)}{(1-\gamma)^2 d(A,r')}$ on the supporting line of r', and one lateral side on q. The homothety with center O and ratio γ^{-1} transforms T' into the trapezoid $T := O + \gamma^{-1}(T' - O)$. Since the area of T' is $3(1-\gamma)^{-2}\,\mathrm{smc}(R)$, the area of T is $3(1-\gamma)^{-2}\gamma^{-2}\,\mathrm{smc}(R)$. We show that $\mathfrak{V}_2(A) \subseteq T$. See Figure 3 for an illustration.

Let B be a point of $\mathfrak{V}_2(A)$. We use $A'B'$ to denote the base segment of $\overline{AB}$ such that $|AA'| < |AB'|$. By the definition of $\mathfrak{V}_2(A)$, we have $|A'B'| \geqslant \gamma|AB|$, $A \notin \mathrm{Vis}(r'', R)$, and $d(B, r'') \leqslant d(A, r'')$, where r'' denotes the root of the base body of $\overline{AB}$. By Lemma 12 and the fact that $A \notin \mathrm{Vis}(r'', R)$, we have $r' = r''$. The bound $d(A, r') \geqslant d(B, r')$ thus implies $A' \in r \cap p^+$ and $B \in q^+$. We have $d(C, r') = d(C) \geqslant 0$ for every $C \in A'B'$.

Observe that $(1-\gamma)d(A, r') \leqslant d(A', r') \leqslant d(A, r')$. The upper bound is trivial, as $d(B, r') \leqslant d(A, r')$ and A' lies on $\overline{AB}$. For the lower bound, we use the expression $A' = tA + (1-t)B'$ for some $t \in [0, 1]$. This gives us $d(A', r') = td(A, r') + (1-t)d(B', r')$. By the estimate $|A'B'| \geqslant \gamma|AB|$, we have

$$|AA'| + |BB'| \leqslant (1-\gamma)|AB| = (1-\gamma)(|AB'| + |BB'|).$$

This can be rewritten as $|AA'| \leqslant (1-\gamma)|AB'| - \gamma|BB'|$. Consequently, $|BB'| \geqslant 0$ and $\gamma > 0$ imply $|AA'| \leqslant (1-\gamma)|AB'|$. This implies $t \geqslant 1-\gamma$. Applying the bound $d(B', r') \geqslant 0$, we conclude that $d(A', r') \geqslant (1-\gamma)d(A, r')$.

Let $(G_n)_{n\in\mathbb{N}}$ be a sequence of points from $A'B'$ that converges to A'. For every $n \in \mathbb{N}$, there is a point $H_n \in r'$ such that $\overline{G_nH_n} \subseteq R$. Since $\overline{r'}$ is compact, there is a subsequence of $(H_n)_{n\in\mathbb{N}}$ that converges to a point $H_0 \in \overline{r'}$. We claim that $H_0 \in q$. Suppose otherwise, and

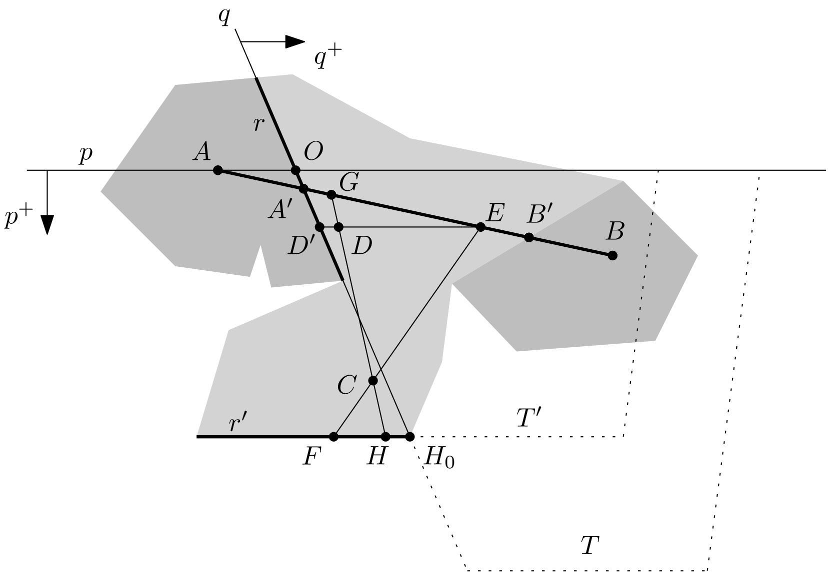


Figure 4 Illustration for the proof of Claim 2 in the proof of Lemma 13.

let $q' \neq q$ be the supporting line of $\overline{A'H_0}$. Let $\varepsilon > 0$ be small enough so that $\mathcal{N}_\varepsilon(A') \subseteq R$. For n large enough, $\overline{G_nH_n}$ is contained in an arbitrarily small neighborhood of q'. Consequently, for n large enough, the supporting line of $\overline{G_nH_n}$ intersects q at a point K_n such that $\overline{G_nK_n} \subseteq \mathcal{N}_\varepsilon(A')$, which implies $K_n \in r \cap \operatorname{Vis}(r', R)$, a contradiction.

Again, let $\varepsilon > 0$. There is a point $E \in A'B'$ such that $|B'E| < \varepsilon$. Let D' be a point of q with $d(D', r') = d(E)$. Let $\delta > 0$. There are points $G \in A'B'$ and $H \in r'$ such that $G \in \mathcal{N}_\delta(A')$ and $\overline{GH} \subseteq R \cap \mathcal{N}_\delta(q)$. If δ is small enough, then $d(E) \leqslant d(A', r') - \delta \leqslant d(G) \leqslant d(A', r')$. Let D be the point of $\overline{GH}$ with $d(D) = d(E)$. The point E lies on $A'B'$ and thus it is visible from a point $F \in r'$. Again, we can choose F so that the line segments $\overline{EF}$ and $\overline{GH}$ have a point C in common where $C \neq F, H$. The points E, F, H, G form a self-intersecting tetragon Q whose boundary is in R. The interior of Q is contained in R, as R is simply connected. Therefore, the area of the triangles CEG and CFH is at most $\operatorname{smc}(R)$. The argument used in the proof of Claim 1 yields $|DE| \leqslant \frac{2\operatorname{smc}(R)}{d(G)^2}(d(G)+d(E)) \leqslant \frac{2\operatorname{smc}(R)}{(d(A',r')-\delta)^2}(d(A', r')+d(E))$. This and the fact that δ (and consequently $|D'D|$) can be made arbitrarily small yield $|D'E| \leqslant \frac{2\operatorname{smc}(R)}{d(A',r')^2}(d(A', r') + d(E))$. This together with $d(A', r') \geqslant (1-\gamma)d(A, r')$ yield $|D'E| \leqslant \frac{2\operatorname{smc}(R)}{(1-\gamma)^2 d(A,r')^2}(d(A, r') + d(E))$. This and $0 \leqslant d(E) \leqslant d(A, r')$ imply $E \in T'$. Since ε can be made arbitrarily small and T' is compact, we have $B' \in T'$. Since $|A'B'| \geqslant \gamma|AB| \geqslant \gamma|A'B|$, we conclude that $B \in T$. This completes the proof of Claim 2.

▶ Claim 3. *$\mathfrak{V}_3(A)$ is contained in a trapezoid $\mathfrak{T}_3(A)$ with area* $(4(1-\gamma)^{-2} - 1)\operatorname{smc}(R)$.

By Lemma 9, the points of r that are visible from A in R form a subsegment CD of r. The homothety with center A and ratio $2(1-\gamma)^{-1}$ transforms the triagle $T' := ACD$ into the triangle $T'' := A + 2(1-\gamma)^{-1}(T' - A)$. See Figure 5 for an illustration. We claim that $\mathfrak{V}_3(A)$ is a subset of the trapezoid $T := T'' \setminus T'$.

Let B be an arbitrary point of $\mathfrak{V}_3(A)$. Consider the segment $\overline{AB}$ with the base segment $A'B'$ such that $|AA'| < |AB'|$. Since $B \in \mathfrak{V}_3(A)$, we have $|A'B'| < \gamma|AB|$ and $|AA'| \geqslant |BB'|$. This implies $|AA'| \geqslant \frac{1-\gamma}{2}|AB| > 0$ and hence $A \neq A'$ and $B \notin P$. From the definition of C and D, we have $A' \in \overline{CD}$. Since $|AA'| \geqslant \frac{1-\gamma}{2}|AB|$ and $B \notin P$, we have $B \in T$.

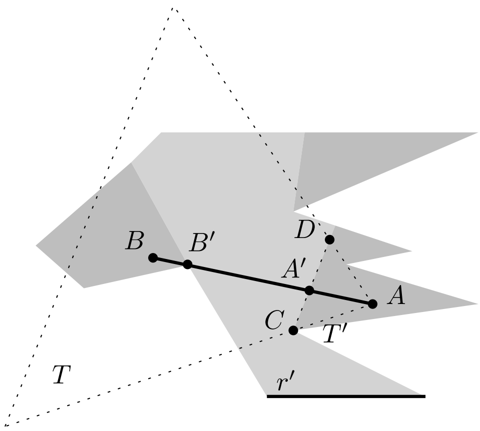


Figure 5 Illustration for the proof of Claim 3 in the proof of Lemma 13.

The area of T is $(4(1-\gamma)^{-2}-1)\lambda_2(T')$. The interior of T' is contained in R, as all points of the open segment CD are visible from A in R. The area of T' is at most $\operatorname{smc}(R)$, as its interior is a convex subset of R. Consequently, the area of T is at most $(4(1-\gamma)^{-2}-1)\operatorname{smc}(R)$. This completes the proof of Claim 3.

To put everything together, we set $\mathfrak{T}(A) := \bigcup_{i=1}^{3} \mathfrak{T}_i(A)$. It follows that $\bigcup_{i=1}^{3} \mathfrak{V}_i(A) \subseteq \mathfrak{T}(A)$ for every $A \in R$. Clearly, the set $\mathfrak{T}(A)$ is measurable. Summing the three estimates on areas of the trapezoids, we obtain

$$\lambda_2(\mathfrak{T}(A)) \leqslant \big(6\gamma^{-2} + 3(1-\gamma)^{-2}\gamma^{-2} + 4(1-\gamma)^{-2} - 1\big)\operatorname{smc}(R)$$

for every point $A \in R$. We choose $\gamma \in (0,1)$ so that the value of the coefficient is minimized. For $x \in (0,1)$, the function $x \mapsto 6x^{-2} + 3(1-x)^{-2}x^{-2} + 4(1-x)^{-2} - 1$ attains its minimum $86.7027 < 87$ at $x \approx 0.5186$. Altogether, we have $\lambda_2(\mathfrak{T}(A)) < 87\operatorname{smc}(R)$ for every $A \in R$.

It remains to show that the set $\{(A,B)\colon A \in R \text{ and } B \in \mathfrak{T}(A)\}$ is measurable. For every body P of R and for $i \in \{1,2,3\}$, the definition of the trapezoid $\mathfrak{T}_i(A)$ in Claim i implies that the set $\{(A,B)\colon A \in P \text{ and } B \in \mathfrak{T}_i(A)\}$ is the intersection of $P \times \mathbb{R}^2$ with a semialgebraic (hence measurable) subset of $(\mathbb{R}^2)^2$ and hence is measurable. There are countably many bodies of R, as each of them has positive measure. Therefore, $\{(A,B)\colon A \in R \text{ and } B \in \mathfrak{T}(A)\}$ is a countable union of measurable sets and hence is measurable. ◀

Let S be a bounded open subset of the plane, and let ℓ be a diagonal of S that lies on the x-axis. For a point $A \in S$, we define the set

$$\mathfrak{S}(A,\ell) := \{B \in \operatorname{Vis}(A,S)\colon AB \cap \ell \neq \emptyset \text{ and } |y(A)| \geqslant |y(B)|\}.$$

The following lemma is a slightly more general version of a result of Cabello et al. [7].

▶ **Lemma 14.** *Let S be a bounded open simply connected subset of $\mathbb{R}^2$, and let ℓ be its diagonal that lies on the x-axis. It follows that $\lambda_2(\mathfrak{S}(A,\ell)) \leqslant 3\operatorname{smc}(S)$ for every $A \in S$.*

Proof. Using an argument similar to the proof of Lemma 8, we can show that the set $\{B \in \operatorname{Vis}(A,S)\colon AB \cap \ell \neq \emptyset\}$ is open. Therefore, $\mathfrak{S}(A,\ell)$ is the intersection of an open set and the closed half-plane $\{(x,y) \in \mathbb{R}^2\colon y \leqslant -y(A)\}$ or $\{(x,y) \in \mathbb{R}^2\colon y \geqslant -y(A)\}$, whichever contains A. Consequently, the set $\mathfrak{S}(A,\ell)$ is measurable for every point $A \in S$.

We clearly have $\lambda_2(\mathfrak{S}(A,\ell)) = 0$ for points $A \in S \smallsetminus \operatorname{Vis}(\ell,S)$. By Lemma 9, the set $\operatorname{Vis}(A,S) \cap \ell$ is an open subsegment CD of ℓ. The interior T° of the triangle $T := ACD$ is contained in S. Since T° is a convex subset of S, we have $\lambda_2(T^\circ) = \frac{1}{2}|CD| \cdot |y(A)| \leqslant \mathrm{smc}(S)$. Therefore, every point $B \in \mathfrak{S}(A,\ell)$ is contained in a trapezoid of height $|y(A)|$ with bases of length $|CD|$ and $2|CD|$. The area of this trapezoid is $\frac{3}{2}|CD| \cdot |y(A)| \leqslant 3\,\mathrm{smc}(S)$. Hence we have $\lambda_2(\mathfrak{S}(A,\ell)) \leqslant 3\,\mathrm{smc}(S)$ for every point $A \in S$. ◀

Proof of Theorem 3. In view of Lemma 7, we can assume without loss of generality that S is an open bounded simply connected set. Let ℓ be a diagonal of S. We can assume without loss of generality that ℓ lies on the x-axis. According to Lemma 10, the set $S \setminus \ell$ has exactly two p-components S_1 and S_2, the sets $S_1 \cup \ell$ and $S_2 \cup \ell$ are rooted sets, and ℓ is their common root. By Lemma 13, for $i \in \{1,2\}$, every point $A \in S_i \cup \ell$ can be assigned a measurable set $\mathfrak{T}_i(A)$ so that $\lambda_2(\mathfrak{T}_i(A)) < 87\,\mathrm{smc}(S_i \cup \ell) \leqslant 87\,\mathrm{smc}(S)$, every line segment $\overline{BC}$ in $S_i \cup \ell$ satisfies $B \in \mathfrak{T}_i(C)$ or $C \in \mathfrak{T}_i(B)$, and the set $\{(A,B)\colon A \in S_i \cup \ell \text{ and } B \in \mathfrak{T}_i(A)\}$ is measurable. We set $\mathfrak{S}(A) := \mathfrak{T}_i(A) \cup \mathfrak{S}(A,\ell)$ for every point $A \in S_i$ with $i \in \{1,2\}$. We set $\mathfrak{S}(A) := \mathfrak{T}_1(A) \cup \mathfrak{T}_2(A)$ for every point $A \in \ell = S \setminus (S_1 \cup S_2)$. Let

$$\mathfrak{S} := \{(A,B)\colon A \in S \text{ and } B \in \mathfrak{S}(A)\} \cup \{(B,A)\colon A \in S \text{ and } B \in \mathfrak{S}(A)\} \subseteq (\mathbb{R}^2)^2.$$

It follows that the set $\mathfrak{S}$ is measurable.

Let $\overline{AB}$ be a line segment in S, and suppose $|y(A)| \geqslant |y(B)|$. Then either A and B are in distinct p-components of $S \setminus \ell$ or they both lie in the same component S_i with $i \in \{1,2\}$. In the first case, we have $B \in \mathfrak{S}(A)$, since AB intersects ℓ and $\mathfrak{S}(A,\ell) \subseteq \mathfrak{S}(A)$. In the second case, we have $B \in \mathfrak{T}_i(A) \subseteq \mathfrak{S}(A)$ or $A \in \mathfrak{T}_i(B) \subseteq \mathfrak{S}(B)$. Therefore, we have $\mathrm{Seg}(S) \subseteq \mathfrak{S}$. Since both $\mathrm{Seg}(S)$ and $\mathfrak{S}$ are measurable, we have

$$\lambda_4(\mathrm{Seg}(S)) \leqslant \lambda_4(\mathfrak{S}) \leqslant 2\int_{A \in S} \lambda_2(\mathfrak{S}(A)),$$

where the second inequality is implied by Fubini's Theorem. Using the bound $\lambda_2(\mathfrak{S}(A)) \leqslant 90\,\mathrm{smc}(S)$, we obtain

$$\lambda_4(\mathrm{Seg}(S)) \leqslant 2\int_S 90\,\mathrm{smc}(S) = 180\,\mathrm{smc}(S)\lambda_2(S).$$

Finally, this bound can be rewritten as $\mathrm{b}(S) = \lambda_4(\mathrm{Seg}(S))\lambda_2(S)^{-2} \leqslant 180\,\mathrm{c}(S)$. ◀

3 General dimension

In this section, we sketch the proofs of Theorem 5 and Theorem 6. The detailed proofs can be found in the full version of this paper [1]. In both proofs, we use the operator Aff to denote the affine hull of a set of points.

Sketch of the proof of Theorem 5. Let $T = (B_0, B_1, \ldots, B_d)$ be a $(d+1)$-tuple of distinct affinely independent points of S, ordered in such a way that the following two conditions hold:

1. the segment $\overline{B_0B_1}$ is the diameter of T, and
2. for $i = 2, \ldots, d-1$, the point B_i has the maximum distance to $\mathrm{Aff}(\{B_0, \ldots, B_{i-1}\})$ among the points $B_i, B_{i+1}, \ldots, B_d$.

For $i = 1, \ldots, d-1$, we define $\mathrm{Box}_i(T)$ inductively as follows:

1. $\mathrm{Box}_1(T) := \overline{B_0B_1}$,
2. for $i = 2, \ldots, d-1$, $\mathrm{Box}_i(T)$ is the box containing all the points $P \in \mathrm{Aff}(\{B_0, B_1, \ldots, B_i\})$ with the following two properties:

  a. the orthogonal projection of P to $\mathrm{Aff}(\{B_0, B_1, \dots, B_{i-1}\})$ lies in $\mathrm{Box}_{i-1}(T)$, and
  b. the distance of P to $\mathrm{Aff}(\{B_0, B_1, \dots, B_{i-1}\})$ does not exceed the distance of B_i to $\mathrm{Aff}(\{B_0, B_1, \dots, B_{i-1}\})$,

3. $\mathrm{Box}_d(T)$ is the box containing all the points $P \in \mathbb{R}^d$ such that the orthogonal projection of P to $\mathrm{Aff}(\{B_0, B_1, \dots, B_{d-1}\})$ lies in $\mathrm{Box}_{d-1}(T)$ and $\lambda_d(\mathrm{Conv}(\{B_0, B_1, \dots, B_{d-1}, P\})) \leqslant \lambda_d(S)\,\mathrm{c}(S)$.

It can be verified that if $T \in \mathrm{Simp}_d(S)$, then $\mathrm{Box}_d(T)$ contains the point B_d. Also, it can be shown that the λ_d-measure of $\mathrm{Box}_d(T)$ is equal to $z := 2^{d-2}d!\,\mathrm{smc}(S)$, which is independent of T. From this, we can deduce that the measure of $\mathrm{Simp}_d(S)$ is at most $(d+1)\lambda_d(S)^d z$, and hence $\mathrm{b}_d(S)$ is at most $(d+1)z/\lambda_d(S)$, which is of order $\mathrm{c}(S)$. ◀

Sketch of the proof of Theorem 6. To obtain a set S with arbitrarily small convexity ratio $\mathrm{c}(S)$ and with the d-index of convexity $\mathrm{b}_d(S)$ of order $\mathrm{c}(S)/\log_2(1/\mathrm{c}(S))$, we let S be the open d-dimensional box $(0,1)^d$ with n points removed. We show that no matter which n-tuple of points we remove from the box, the d-index of convexity $\mathrm{b}_d(S)$ is still of order $\Omega(\frac{1}{n})$. Moreover, we show that for some constant $\alpha = \alpha(d) > 0$ it is possible to remove $n = \alpha\frac{1}{\varepsilon}\log_2\frac{1}{\varepsilon}$ points from the box such that every convex subset of $(0,1)^d$ with measure at least ε contains a removed point. That is, we obtain $\mathrm{c}(S) \leqslant \varepsilon$ and $\mathrm{b}_d(S) \geqslant \gamma\varepsilon/\log_2(1/\varepsilon)$ for some constant $\gamma = \gamma(d) > 0$. Such an n-tuple of points to be removed is called an *ε-net* for convex subsets of $(0,1)^d$. To find it, we first use John's Lemma [11] to reduce the problem to finding, for a suitably smaller ε', an *ε'-net* for ellipsoids restricted to $(0,1)^d$. Then, we apply a continuous version of the well-known Epsilon Net Theorem for families with bounded Vapnik-Chervonenkis dimension due to Haussler and Welzl [10] (see also [14]). ◀

It is a natural question whether the bound for $\mathrm{b}_d(S)$ in Theorem 6 can be improved to $\mathrm{b}_d(S) = \Omega(\mathrm{c}(S))$. In the plane, this is related to the famous problem of Danzer and Rogers (see [6, 15] and Problem E14 in [8]) which asks whether for given $\varepsilon > 0$ there is a set $N' \subseteq (0,1)^2$ of size $O(\frac{1}{\varepsilon})$ with the property that every convex set of area ε within the unit square contains at least one point from N'.

If this problem was to be answered affirmatively, then we could use such a set N' to stab $(0,1)^2$ in our proof of Theorem 6 which would yield the desired bound for $\mathrm{b}_2(S)$. However it is generally believed that the answer is likely to be nonlinear in $\frac{1}{\varepsilon}$.

4 Other variants and open problems

We have seen in Theorem 3 that a p-componentwise simply connected set $S \subseteq \mathbb{R}^2$ whose $\mathrm{b}(S)$ is defined satisfies $\mathrm{b}(S) \leqslant \alpha\,\mathrm{c}(S)$, for an absolute constant $\alpha \leqslant 180$. Equivalently, such a set S satisfies $\mathrm{smc}(S) \geqslant \mathrm{b}(S)\lambda_2(S)/180$.

By a result of Blaschke [5] (see also Sas [18]), every convex set $K \subseteq \mathbb{R}^2$ contains a triangle of measure at least $\frac{3\sqrt{3}}{4\pi}\lambda_2(K)$. In view of this, Theorem 3 yields the following consequence.

▶ **Corollary 15.** *There is a constant $\alpha > 0$ such that every p-componentwise simply connected set $S \subseteq \mathbb{R}^2$ whose $\mathrm{b}(S)$ is defined contains a triangle $T \subseteq S$ of measure at least $\alpha\,\mathrm{b}(S)\lambda_2(S)$.*

A similar argument works in higher dimensions as well. For every $d \geqslant 2$, there is a constant $\beta = \beta(d)$ such that every convex set $K \subseteq \mathbb{R}^d$ contains a simplex of measure at least $\beta\lambda_d(K)$ (see e.g. Lassak [13]). Therefore, Theorem 5 can be rephrased in the following equivalent form.

▶ **Corollary 16.** *For every $d \geqslant 2$, there is a constant $\alpha = \alpha(d) > 0$ such that every set $S \subseteq \mathbb{R}^d$ whose $\mathrm{b}_d(S)$ is defined contains a simplex T of measure at least $\alpha\,\mathrm{b}_d(S)\lambda_d(S)$.*

What can we say about sets $S \subseteq \mathbb{R}^2$ that are not p-componentwise simply connected? First of all, we can consider a weaker form of simple connectivity: we call a set S *p-componentwise simply $\triangle$-connected* if for every triangle T such that $\partial T \subseteq S$ we have $T \subseteq S$. We conjecture that Theorem 3 can be extended to p-componentwise simply $\triangle$-connected sets.

▶ **Conjecture 17.** *There is an absolute constant $\alpha > 0$ such that every p-componentwise simply $\triangle$-connected set $S \subseteq \mathbb{R}^2$ whose $\mathrm{b}(S)$ is defined satisfies $\mathrm{b}(S) \leqslant \alpha\,\mathrm{c}(S)$.*

What does the value of $\mathrm{b}(S)$ say about a planar set S that does not satisfy even a weak form of simple connectivity? Such a set may not contain any convex subset of positive measure, even when $\mathrm{b}(S)$ is equal to 1. However, we conjecture that a large $\mathrm{b}(S)$ implies the existence of a large convex set whose boundary belongs to S.

▶ **Conjecture 18.** *For every $\varepsilon > 0$, there is a $\delta > 0$ such that if $S \subseteq \mathbb{R}^2$ is a set with $\mathrm{b}(S) \geqslant \varepsilon$, then there is a bounded convex set $C \subseteq \mathbb{R}^2$ with $\lambda(C) \geqslant \delta\lambda(S)$ and $\partial C \subseteq S$.*

Theorem 3 shows that Conjecture 18 holds for p-componentwise simply connected sets, with δ being a constant multiple of ε. It is possible that even in the general setting of Conjecture 18, δ can be taken as a constant multiple of ε.

Motivated by Corollary 15, we propose a stronger version of Conjecture 18, where the convex set C is required to be a triangle.

▶ **Conjecture 19.** *For every $\varepsilon > 0$, there is a $\delta > 0$ such that if $S \subseteq \mathbb{R}^2$ is a set with $\mathrm{b}(S) \geqslant \varepsilon$, then there is a triangle $T \subseteq \mathbb{R}^2$ with $\lambda(T) \geqslant \delta\lambda(S)$ and $\partial T \subseteq S$.*

Note that Conjecture 19 holds when restricted to p-componentwise simply connected sets, as implied by Corollary 15.

We can generalise Conjecture 19 to higher dimensions and to higher-order indices of convexity. To state the general conjecture, we introduce the following notation: for a set $X \subseteq \mathbb{R}^d$, let $\binom{X}{k}$ be the set of k-element subsets of X, and let the set $\mathrm{Skel}_k(X)$ be defined by

$$\mathrm{Skel}_k(X) := \bigcup_{Y \in \binom{X}{k+1}} \mathrm{Conv}(Y).$$

If X is the vertex set of a d-dimensional simplex $T = \mathrm{Conv}(X)$, then $\mathrm{Skel}_k(X)$ is often called the *k-dimensional skeleton* of T. Our general conjecture states, roughly speaking, that sets with large k-index of convexity should contain the k-dimensional skeleton of a large simplex. Here is the precise statement.

▶ **Conjecture 20.** *For every $k, d \in \mathbb{N}$ such that $1 \leqslant k \leqslant d$ and every $\varepsilon > 0$, there is a $\delta > 0$ such that if $S \subseteq \mathbb{R}^d$ is a set with $\mathrm{b}_k(S) \geqslant \varepsilon$, then there is a simplex T with vertex set X such that $\lambda_d(T) \geqslant \delta\lambda_d(S)$ and $\mathrm{Skel}_k(X) \subseteq S$.*

Corollary 16 asserts that this conjecture holds in the special case of $k = d \geqslant 2$, since $\mathrm{Skel}_d(X) = \mathrm{Conv}(X) = T$. Corollary 15 shows that the conjecture holds for $k = 1$ and $d = 2$ if S is further assumed to be p-componentwise simply connected. In all these cases, δ can be taken as a constant multiple of ε, with the constant depending on k and d.

Finally, we can ask whether there is a way to generalize Theorem 3 to higher dimensions, by replacing simple connectivity with another topological property. Here is an example of one such possible generalization.

▶ **Conjecture 21.** *For every $d \geqslant 2$, there is a constant $\alpha = \alpha(d) > 0$ such that if $S \subseteq \mathbb{R}^d$ is a set with defined $\mathrm{b}_{d-1}(S)$ whose every p-component is contractible, then $\mathrm{b}_{d-1}(S) \leqslant \alpha\, \mathrm{c}(S)$.*

A modification of the proof of Theorem 5 implies that Conjecture 21 is true for star-shaped sets S.

Acknowledgment. The authors would like to thank to Marek Eliáš for interesting discussions about the problem and participation in our meetings during the early stages of the research.

References

1. M. Balko, V. Jelínek, P. Valtr, and B. Walczak. On the Beer index of convexity and its variants. full version, arXiv:1412.1769.
2. G. Beer. Continuity properties of the visibility function. *Michigan Math. J.*, 20:297–302, 1973.
3. G. Beer. The index of convexity and the visibility function. *Pacific J. Math.*, 44(1):59–67, 1973.
4. G. Beer. The index of convexity and parallel bodies. *Pacific J. Math.*, 53(2):337–345, 1974.
5. W. Blaschke. Über affine Geometrie III: Eine Minimumeigenschaft der Ellipse. *Ber. Verh. Kön. Sächs. Ges. Wiss. Leipzig Math.-Phys. Kl.*, 69:3–12, 1917.
6. P. G. Bradford and V. Capoyleas. Weak ε-nets for points on a hypersphere. *Discrete Comput. Geom.*, 18(1):83–91, 1997.
7. S. Cabello, J. Cibulka, J. Kynčl, M. Saumell, and P. Valtr. Peeling potatoes near-optimally in near-linear time. In *Proceedings of the 30th Annual Symposium on Computational Geometry*, pages 224–231, 2014.
8. H. T. Croft, K. J. Falconer, and R. K. Guy. *Unsolved Problems in Geometry.* Unsolved Problems in Intuitive Mathematics. Springer New York, 2nd edition, 1991.
9. J. E. Goodman. On the largest convex polygon contained in a non-convex n-gon, or how to peel a potato. *Geom. Dedicata*, 11(1):99–106, 1981.
10. D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2(2):127–151, 1987.
11. F. John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays, presented to R. Courant on his 60th birthday, January 8, 1948*, pages 187–204, 1948.
12. R. Lang. A note on the measurability of convex sets. *Arch. Math. (Basel)*, 47:90–92, 1986.
13. M. Lassak. Approximation of convex bodies by inscribed simplices of maximum volume. *Beitr. Algebra Geom.*, 52(2):389–394, 2011.
14. J. Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics.* Springer New York, 2002.
15. J. Pach and G. Tardos. Piercing quasi-rectangles—on a problem of Danzer and Rogers. *J. Combin. Theory Ser. A*, 119(7):1391–1397, 2012.
16. V. V. Prasolov. *Elements of combinatorial and differential topology*, volume 74 of *Graduate Studies in Mathematics.* American Mathematical Society, 2006.
17. G. Rote. The degree of convexity. In *Abstracts of the 29th European Workshop on Computational Geometry*, pages 69–72, 2013.
18. E. Sas. Über eine Extremumeigenschaft der Ellipsen. *Compositio Math.*, 6:468–470, 1939.
19. H. I. Stern. Polygonal entropy: a convexity measure for polygons. *Pattern Recogn. Lett.*, 10(4):229–235, 1989.