Algorithms and Combinatorics for Beyond Planar Graphs

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Maximilian Pfister

aus Albstadt-Ebingen

Tübingen

2023

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	08.05.2024
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Michael Kaufmann
2. Berichterstatter/-in:	Prof. Dr. Klaus-Jörn Lange
3. Berichterstatter/-in:	Prof. Dr. Markus Chimani

"It's fine to work on any problem, so long as it generates interesting mathematics along the way — even if you don't solve it at the end of the day."

Andrew Wiles

UNIVERSITÄT TÜBINGEN

Abstract

Mathematisch-Naturwissenschaftliche Fakultät
Fachbereich Informatik

Doktor der Naturwissenschaften

Algorithms and Combinatorics for Beyond Planar Graphs

by Maximilian Pfister

Planar graphs form a cornerstone in the area of graph drawing with a rich and diverse literature ranging from combinatorial results to algorithmic applications such as the recognition problem of planar graphs or the existence of a morph between two planar drawings. Unfortunately, these seminal results do not easily extend to most proposed generalizations of planar graphs. In the first part of this thesis, we make progress in this direction and provide (approximation) algorithms to efficiently test if graphs of small pathwidth are k-planar, i.e., test if they can be drawn such that every edge is crossed at most k times. We further describe an algorithm that computes a morph for a meaningful family of 1-planar graphs, which is the first result in this direction for graphs of non-constant genus. We conclude the first part by considering the containment relation of low-degree graphs in the k-bend RAC setting by providing efficient drawing algorithms.

In the second part, we study structural properties of beyond-planar graph classes. Motivated by the results from the previous chapter and inspired by orthogonal drawings, we introduce a subclass of RAC graphs for which we study the recognition problem, edge-density bounds and their relationship to general RAC graphs. We next consider the class of gap-planar graphs and establish a tight upper bound on the edge-density these graphs can obtain. We further consider fan-planar graphs, where we show that all three forbidden patterns which are used to define fan-planar graphs are in fact necessary (i.e., allowing or prohibiting each pattern gives rise to distinct classes of graphs). Whereas previous work showed that this is true for two of the three patterns, we explicitly consider the third pattern which gives rise to weak and strong fan-planar graphs. We establish edge-density bounds for the class of weak fan-planar graphs, thus extending previous results on strong fan-planar ones. We finally study the maximum thickness of strong fan-planar graphs and establish the first (non-trivial) relation between the graph thickness and a beyond-planar graph class.

In the final part of this thesis, we provide an algorithmic framework for the simultaneous optimization of several aesthetic criteria which are characterized to be important in a good drawing. The main contribution of our algorithm is its ease of extension to additional criteria, since no empirically found coefficients or (surrogate) differentiable loss functions are required, which is common for other algorithms that optimize many such criteria in a joint way.

Zusammenfassung

Algorithms and Combinatorics for Beyond Planar Graphs

Planare Graphen bilden einen Eckpfeiler im Bereich der Graphenzeichnung mit einer reichhaltigen und vielfältigen Literatur, welche von kombinatorischen Ergebnissen bis hin zu algorithmischen Anwendungen, wie zum Beispiel dem effizienten Testen der Planarität oder der Berechnung eines Morphs zwischen zwei planaren Zeichnungen, reicht. Leider lassen sich dies grundlegenden Ergebnisse nicht ohne Weiteres auf mögliche Verallgemeinerungen planarer Graphen übertragen. Im ersten Teil der Arbeit befassen wir uns mit dieser Problematik und beschreiben Algorithmen, welche für Graphen mit kleiner Pfadweite effizient testen können, ob diese eine Zeichnung in der Ebene besitzen, in der jede Kante maximal k andere Kanten kreuzt. Außerdem entwicklen wir einen Algorithmus, der einen Morph für eine bedeutende Familie von 1-planaren Graphen berechnet (welches diesbezüglich das erste Resultat für Graphen mit nicht konstantem Geschlecht ist). Desweiteren entwerfen wir Algorithmen, welche für Graphen mit kleinem Grad eine RAC Zeichnung mit wenigen Knicken liefert.

Im zweiten Teil der Arbeit untersuchen wir strukturelle Eigenschaften einiger Graphklassen im Forschungsgebiet "Beyond Planarity". Inspiriert von dem klassichen orthogonalen Zeichenstil führen wir zunächst eine Unterklasse von RAC-Graphen ein, für die wir die maximale Kantendichte, die Beziehung zu allgemeinen RAC-Graphen sowie das zugehörige Erkennungsproblem untersuchen. Im Anschluss zeigen wir eine obere Schranke für die Kantendichte von bipartiten gap-planaren Graphen, welche, bis auf eine konstante Anzahl an Kanten, bestmöglich ist. Außerdem betrachten wir die Klasse der fan-planaren Graphen, welche durch drei verbotene Kreuzungskonfigurationen definiert ist. Wir zeigen, dass das Zulassen oder Verbieten der dritten Konfiguration zu unterschiedlichen Klassen von Graphen führt und führen in diesem Sinne die Unterscheidung zwischen schwachen- und starken fan-planaren Graphen ein. Wir erweitern das Ergebnis der oberen Schranke der Kantendichte von starken zu schwachen fan-planaren Graphen und untersuchen außerdem die maximale Graphendicke von starken fan-planaren Graphen, welches die erste (nicht-triviale) Beziehung zwischen der Graphendicke und einer Graphklasse im Gebiet "Beyond Planarity" darstellt.

Im letzten Teil der Arbeit führen wir ein algorithmisches Framework zur Visualisierung von Graphen ein, welches mehrere ästhetische Kriterien simultan optimieren kann. Der Hauptbeitrag unseres Frameworks besteht darin, dass es leicht auf zusätzliche Kriterien erweitert werden kann, da, im Gegensatz zu vielen anderen Algorithmen dieser Art, keine empirisch gefundenen Koeffizienten oder differenzierbare Zielfunktionen benötigt werden.

Acknowledgements

I would like to thank my supervisor Michael for giving me the opportunity to write this thesis, for his continuous support throughout the years and for the fruitful yet joyful moments we shared while researching together (in particular, the ones very early in the morning).

I would also like to thank the current and former members of the algorithms group: Axel, Henry, Julia, Lena, Michalis, Patri and Thomas for providing support in both research and teaching, for providing an atmosphere where even the dumbest ideas were profoundly discussed and not immediately dismissed - I cherish these years of intellectual exchange a lot. I would also like to thank Renate for being the person in the background that keeps everything running administrative wise.

Further, I would also like to thank my co-authors which are not part of the Tübingen group: Fabri, Ignaz, Miriam, Otfried and Torsten for exchanging interesting ideas and not being annoyed by the omnipresent Zoom calls in these troublesome past years.

Finally, I would like to express my deepest gratitude to my family, my girlfriend and all my friends for their unwavering support throughout the years and for all the joyful diversions from the dreary moments when research did not advance as desired.

Contents

List of Figures

Dedicated to my late father.

Chapter 1

Introduction

Influenced by technical prowess and by scientific breakthroughs, particularly in the last century, our understanding of almost every topic conceivable is ever-increasing. Due to this deeper understanding, we uncover underlying systems that are more and more complex; a prime example being the step from classical mechanics to quantum mechanics. Many such complex systems consist of a vast number of entities that interact with each other and hence they are destined to be modeled as *graphs*. A graph $G = (V, E)$ consists of vertices V (which can be, for example, particles) and edges E, which are tuples of vertices (e.g., it describes the interaction between these particles). The study of graphs forms a branch of mathematics called *graph theory*, whose beginning is attributed to Leonard Euler in 1736 and his paper about the famous "Seven bridges of Königsberg" problem [88][1]. Subsequent research on graphs is vast and concerns many different topics which are applicable to numerous real-world problems, e.g., the electrical distribution of a system can be modeled as a flow problem or the fastest route for a vehicle can be computed using the shortest-path paradigm. The two most common ways for a computer to efficiently store and query graph data is the *adjacency matrix* or the *adjacency list*. While the former has a constant query time, it requires quadratic space in the number of vertices independent of the number of edges - thus for sparse graphs, i.e., graphs that do not contain "many" edges, the latter variant is usually preferred. In Fig. 1.1, we compare different representations of the same abstract graph. Consider the task to find the shortest path from vertex 4 to vertex 7 - it turns out that humans have to invest some effort to obtain the desired result using either the adjacency matrix or the adjacency list, while we (arguably) obtain the fastest result using the *drawing* of the graph.

Adjacency matrix

	1	2	3	4	5	6	7
1	0	1	0	1	1	0	1
2	1	0	1	0	0	0	0
3	0	1	0	1	0	0	0
4	1	0	1	0	1	0	0
5	1	0	0	1	0	1	0
6	0	0	0	0	1	0	1
7	1	0	0	0	0	1	0

Adjacency list

1	$\to 2 \to 4 \to 5 \to 7$
2	$\to 1 \to 3$
3	$\to 2 \to 4$
4	$\to 1 \to 3 \to 5$
5	$\to 1 \to 4 \to 6$
6	$\to 5 \to 7$
7	$\to 1 \to 6$

Drawing of the graph

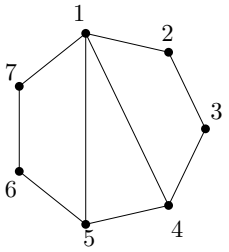


FIGURE 1.1: Three different representations of the same abstract graph.

[1]The problem is whether the graph defined by the set of islands of Königsberg admits an Euler tour, where two islands are connected if there exists a bridge between them

In a sense, this is what the field of *graph drawing* is all about - to create visualizations of graphs which are easy to read and understand for humans. Arguably the first paper in this area dates back to 1963 [134], when Donald Knuth studied the automatic visualization of flow charts which arise when describing computer programs. Automatic visualization of graphs can nowadays be found in a plethora of areas such as software engineering, databases, in real-time systems including the specification of Petri-nets or in chemistry to model chemical bonds in complex molecules [69]. The most common style to draw graphs in the plane are *node-link-diagrams*, where the vertices are represented by symbols and the edges are Jordan arcs connecting these symbols. The drawing of our previous example had the property that no two arcs shared a common interior point, that is, a *crossing*. A drawing without any crossing is called a *planar* drawing and a graph which admits such a drawing is called a *planar* graph. One of the earliest results for planar graphs dates back to Euler [89], who extended the formula $|V(G)| - |E(G)| + |F(G)| = 2$, which relates the number of vertices $|V(G)|$, edges $|E(G)|$ and faces[2] $|F(G)|$, from convex polyhedra to (connected) planar graphs. Euler also established that $|E(G)| \leq 3|V(G)| - 6$ has to hold for a graph to be planar. We refer the interested reader to [121] for a detailed history of planar graphs.

Since the following part of the introduction will delve into more technical details, we refer the reader to Chapter 2 for definitions.

Natural extensions beyond planar graphs The earlier mentioned condition on the maximum number of edges that a planar graph can have gives rise to the following question: Can we somehow generalize the notion of planarity to graphs that have (slightly) more than $3|V(G)| - 6$ edges while maintaining some of the desired properties of planar graphs? To this end, let us consider the following possible extensions:

1. A planar subgraph G' of G can be obtained by removing at most k vertices or k edges from G.
2. G admits an embedding on a surface of genus k.
3. G admits a drawing in the plane with k crossings.
4. G has *thickness* $(k+1)$, i.e., the edges of G can be partitioned into $(k+1)$ disjoint sets such that the subgraphs induced by each set is planar.
5. G admits a drawing in the plane where every edge is crossed at most k times.

Clearly, a planar graph satisfies all of these properties with $k = 0$.

In the following, we will consider three of the most important topics that were studied in the context of planar graphs, namely the *recognition* of planar graphs, the *morphing* of planar drawings and the restriction to *straight-line* planar drawings. For each of them, we will first consider seminal results for planar graphs, give an overview of the related work for the aforementioned extensions and then describe our contribution to these topics, which form the first part of the thesis.

Recognition of planar graphs The first seminal result dates back to 1930, when Kuratowski's Theorem [138] established that a graph is planar if and only if it does not contain a subgraph H which is a subdivision of K_5 or $K_{3,3}$. Wagner's Theorem [179]

[2] faces are bounded regions of the plane delimited by edges and vertices

states that a graph is planar if and only if the graph does not contain a K_5 or a $K_{3,3}$ *minor*. Since any subdivision H can always be converted into a minor H' by contracting the edges along a subdivision path, while the converse is not always true [117], it seems like the result of Kuratowski is stronger and the "only-if" direction of Wagner might be incorrect - however, it can be shown that if a graph contains a minor of K_5 or $K_{3,3}$, it also contains a subdivision of (one of) K_5 or $K_{3,3}$ [41] which makes the two results equivalent. A generalization of Wagner's Theorem is the seminal result known as the Robertson-Seymour Theorem[3] [157], which establishes that any family of graphs that is closed under minors can be defined by a *finite* set of forbidden minors (i.e., K_5 and $K_{3,3}$ for the case of planar graphs). Moreover, they showed that for graphs H and G, one can verify if H is a minor of G in $f(H)|V(G)|^3$ time [158] for a computable function f. Combining these two results, we obtain that we can test in $\mathcal{O}(|V(G)|^3)$ time if a graph G belongs to a minor-closed family or not. A small caveat here is the hidden constant: since the number of forbidden minors of a family $\mathcal{F}$ only depends on $\mathcal{F}$, it is treated as a constant $c_{\mathcal{F}}$ with respect to the input graph G, but it can be quite large. Further, the algorithm iterates through the set of forbidden minors and tests if the current minor H_i is a minor of the input graph G - here the costs can be expressed as $f(H_i)|V(G)|^3$. Again, $f(H_i)$ does not depend on the input and is treated as a constant, but this can be quite expensive as well. Thus, even though necessary and sufficient conditions for a graph to be planar were established, there was a lot of subsequent research conducted to search for algorithms which allow efficient planarity testing. In 1974, an asymptotically optimal algorithm for planarity testing was found by Hopcroft and Tarjan [115] which verifies in $\mathcal{O}(|V(G)|)$ time if a graph G is planar.

Let us take a step back and consider the recognition problem for the extensions that we proposed earlier.

1. For a fixed k, a simple brute-force algorithm with runtime $\mathcal{O}(|V(G)|^{k+1})$ can verify if a planar subgraph can be obtained from input graph G after removing at most k vertices. Similarly, an algorithm with runtime $\mathcal{O}(|V(G)|^{2k+1})$ exists for the removal of the edges. If k is part of the input, the problem is NP-hard for both vertices [141] and edges [183].

2. Since the family of graphs of genus k for a fixed k is minor-closed (the proof is a straight-forward generalization of the one for planar graphs, i.e., graphs of genus zero), it follows that one can test in polynomial time whether the genus of a graph is k or not. However, to decide for an arbitrary k and a given graph G if the genus of G is at most k is shown to be NP-complete [173].

3. For fixed k, we can again verify if G admits a drawing in the plane with at most k crossings in polynomial time. Namely, for any of the k crossings, we consider all possible pairs of edges which can form this crossing. We substitute the chosen pair of crossing edges with a $K_{1,4}$, i.e., a degree four vertex connected to the endpoints of the pair of crossing edges [4]. Let G' be the resulting graph obtained by applying this procedure for any of the k crossings. If G' is planar (which can be tested in linear time), then $cr(G) \leq k$. The total runtime for this trivial algorithm is $\mathcal{O}(|V(G)|^{4k+1})$; an optimal linear time algorithm can be found in [131]. We remark here that, contrary to planar graphs, the family of graphs with crossing number k for $k \geq 1$ is not minor-closed [103]. Once again, the general problem of determining the crossing number of a given

[3] The work to prove this Theorem spawns over 20 papers and we only cite the last one here

[4] one can show that a crossing minimal drawing of G is simple

graph is NP-complete [104] if k is provided as part of the input. Moreover, it was shown [50] that the corresponding minimization problem is APX-hard, i.e, there exists a constant $c_0 > 1$ such that no (polynomial-time) c_0-approximation of the crossing number is possible. While it is possible that a constant approximation exists, the current best bound of order $\mathcal{O}(\sqrt{|V(G)|})$ for bounded-degree graphs [57] which uses highly non-trivial techniques may indicate the opposite.

4. A graph G has *thickness* k if G is the union of at most k planar graphs. Influenced by the previous extensions, one might think that for a fixed k, deciding if a graph G has thickness k should be polynomial-time solvable - however, this turns out to be NP-complete even for the case of $k = 2$ [145]. If k is fixed and the given graph G has treewidth $t \in O(1)$, then one can show that deciding if G has thickness k can be solved in polynomial-time. While this follows directly by a combination of known properties, we could not find this statement explicitly and thus provide it here for completeness. By Courcelle's Theorem [59], we have that if a graph property is definable in monadic second-order logic (MSO), then it can be decided in linear time on graphs of bounded treewidth. For our case, it is more convenient to use MSO_2, which allows to quantify over sets of edges. A statement in MSO_2 can be transformed into a statement of MSO [61] if the considered graph class C is closed under taking subgraphs and every graph in C is sufficiently sparse, i.e., there exists a constant c such that $|E(G)| \leq c|V(G)|$ for any graph G of C. The first condition is guaranteed by the bounded treewidth, while the latter follows from the fact that, for a fixed k, a graph G with thickness k can have at most $k(3|V(G)| - 6)$ edges. Hence, by setting $c = 3k$, the considered class of graphs is sufficiently sparse. It remains to express the property of thickness k in a MSO formula. Here, we leverage the fact that there exists a MSO formular ψ_{planar} which says that a graph is planar [60]. Thus the resulting MSO_2 formula is as follows:

$$\exists X_1, \ldots, \exists X_k (\forall E e \rightarrow X_1 e \vee \cdots \vee X_k e) \wedge \psi_{planar}(X_1) \wedge \cdots \wedge \psi_{planar} X_k \qquad (1.1)$$

Contrary to the crossing number, there exists a 3-approximation of the thickness by computing the arboricity of a graph, which can be done in polynomial time [101].

5. Testing whether G admits a drawing in the plane where every edge has at most k crossings is NP-complete for any $k \geq 1$ [178]. Moreover, the corresponding minimization problem is APX-hard, as the existence of an approximation algorithm with ratio $2 - \epsilon$ for $\epsilon > 0$ would solve the problem in polynomial time for $k = 1$, hence no such algorithm can exist. No approximation algorithms for any k are known.

To this end, we will develop in Chapter 4 exact and approximate algorithms to test if graphs of small pathwidth admit drawings in the plane where every edge is crossed at most k times.

Morphing of planar drawings The study of morphs[5] of planar graphs originated almost a century ago in the work of Steinitz [49], who, fitting to the origin of planar graphs, proved that there exists a morph for 3-dimensional convex polyhedra (which correspond to 3-connected planar graphs [184]). In 1944, Cairns showed

[5] For a definition of the term *morph*, refer to the introductory paragraph of Chapter 5.

that a morph exists for maximal planar graphs [51], i.e., triangulations, which was later extended by Thomassen [171] to general planar graphs, thus settling previous conjectures by Grünbaum and Shepard [109] and by Robinson [159]. Recent results are concerned with the complexity of the morph [4] or try to maintain additional geometric properties of the drawings throughout the morph such as convexity [14, 87, 133, 171], orthogonality [32, 106] or upwardness [64]. While the algorithm of Cairn computes the trajectories of the vertices explicitly, the work of Floater and Gotsman [91], which is based on the famous barycenter method by Tutte [175] and was later extended by Gotsman and Surazhsky [107] to all planar graphs, provides an algorithm that computes the intermediate drawing at any requested time instance. Finally, we highlight that these rather theoretical results also find use in practical applications, e.g., in computer vision, since a morph between two arbitrary shapes can be approximated by a morph between their corresponding meshes, which in turn are just planar grid-graphs and we can use the aforementioned results [6]. For non-planar graphs, we unfortunately do not have such a rich literature. To the best of our knowledge, there is exactly one result by Chambers et. al. [52] which establishes a morph of graphs on the torus, i.e., a surface of genus one[6]. In Chapter 5, we will present the first morphing algorithm for graphs of non-constant genus. In particular, our algorithm computes a morph for a meaningful family of 1-planar graphs.

Straight-line planar drawings The existential proof that every planar graph admits a straight-line drawing was established independently in [100, 180] almost a century ago. Algorithmic results which construct a straight-line (grid) drawing of a graph G with grid-size polynomial in $|V(G)|$ are due to [56, 97, 163]. To this end, it is still an open question whether there exists a set of grid points S (also called *universal point set*) such that $|S| \in o(|V(G)|^2)$. Let us again consider the natural extensions of planar graphs (in particular, the latter three).

Graphs with crossing number larger than three may have arbitrarily many crossings when the edges are restricted to straight-line segments [34].

The straight-line embedding of thickness k graphs was mostly studied for $k = 2$ with the additional requirement that the two planar subgraphs are crossing free in the resulting drawing. Graphs which admit such a drawing have so called *geometric thickness* two. Clearly, geometric thickness k graphs form a subclass of thickness k graphs. The result of [80] establishes a proper containment for the case of $k = 2$ by proving an upper bound on the density of geometric thickness two graphs (which is strictly smaller than $6|V(G)| - 12$)[7].

Finally, there are graphs that admit drawings where every edge is crossed at most once which do not admit straight-line drawings where every edge is crossed at most once [172].

We observe that none of the three proposed extensions of planar graphs behaves well when restricted to the straight-line setting. Hence, let us relax the straight-line requirement for now and allow the edges to consist of *polylines*, i.e., a chain of straight-line segments that touch at *bend points*. Planar graph drawings have also been extensively studied in the presence of bends. Here, a fundamental result was

[6]Note that every graph with crossing number at most one admits an embedding on the torus by definition, so it is also a small result in that direction

[7]In fact the proper inclusion can be shown for many examples, see [36] for a survey

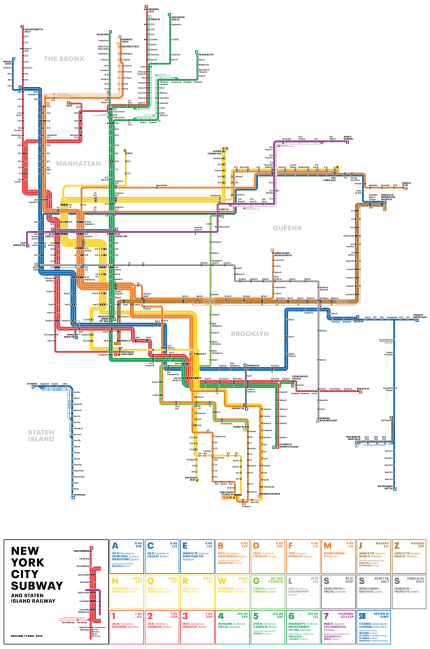


FIGURE 1.2: Orthogonal drawing of the NYC metro map by [177].

established by Tamassia [168] in the context of *orthogonal* graph drawings, i.e., drawings in which edges are axis-aligned polylines. He suggested an approach to minimize the number of bends of degree-4 plane graphs using flows; we refer the interested reader to [71]. The motivation behind axis-aligned polylines is not only of theoretical nature, however. On the one hand, it is used to minimize the area in computer chip design [140], on the other hand, orthogonal drawings are heavily used to visualize graphs from many different fields of applications such as the visualization of metro maps, see Fig. 1.2, which indicates that these drawings meet favorable *aesthetic criteria*.

This brings us back to the original goal of graph drawing and the important question - can we measure the *quality* of a drawing? Most results regarding this question try to identify a set of (measureable) aesthetic criteria that characterize the quality of a drawing. In particular, empirical studies [23, 152, 165] identified several such criteria, e..g., the number of crossings, the area or the symmetry of the drawing. By definition, orthogonal drawings optimize the *angular resolution* of the drawing, which is defined as the minimum angle between two edges that are incident to the same vertex. Since small angles seem to impair the readability of the drawing, orthogonal drawings are best possible (for degree-4 graphs) since they attain exactly 90°. While crossings negatively affect the readability of a drawing [152], we cannot avoid them when drawing non-planar graphs. Fortunately, there is still some hope left to obtain relatively good drawings. The eye-tracking study of [118], where the task was to find the shortest path between two query points, showed that crossings which occur at a large angle only slightly impair the readability of a drawing (i.e., the response time was only slightly slower compared to crossing-free drawings). This implies that the *crossing resolution*, for which we want to maximize the minimum angle formed at a crossing, plays an important role in the quality of a drawing. The reader can verify this observation themselves in Fig. 1.3 - the identification of the path from vertex 1 to vertex 2 is substantially more difficult in the left drawing, which has a small

crossing resolution, compared to the right one. This observation gives rise to the class of right-angle crossing drawings (*RAC* drawings). Similar to orthogonal drawings, which maximize the angular resolution, RAC drawings maximize the crossing resolution as every crossing has to occur at a right angle. In Chapter 6, we study low-degree graphs in the RAC setting both in the presence and absence of bends by providing efficient constructive algorithms. Since the general problem of deciding whether a graph admits a straight-line RAC drawing is NP-hard [16], our results also broaden the subclass of graphs for which this question is trivial.

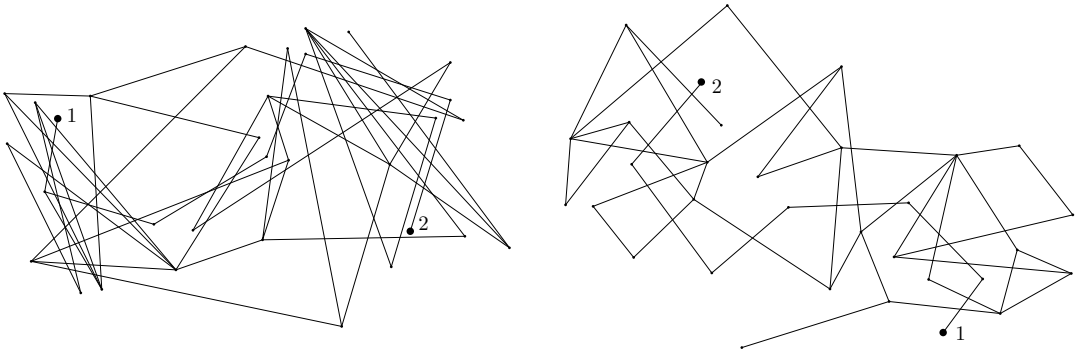


FIGURE 1.3: Example that visualizes the importance of crossing angles, taken from [118]

Beyond planarity Besides the impact of geometric properties of crossings for the readability of a drawing, the experiment conducted in [148] showed that drawings with a higher crossing number but small skewness (in the experiment, the removal of four edges makes the drawing planar) are preferred over crossing-minimal drawings with higer skewness (i.e., more edges have to be removed to obtain a planar drawing). Hence, also *topological* properties of the crossings play a role in determining the quality of a drawing. These observations gave rise to the field of *beyond-planarity*. A graph G belongs to a beyond-planar class C (such as RAC), if G admits a drawing which does not contain a forbidden crossing configuration defined by C - in our example, the forbidden configuration would be of geometric nature, namely a crossing which does not occur at a right angle. For an overview of the topic, consider the recent survey by Didimo et. al. [77].

Recall that the fifth proposed generalization of planar graphs were graphs which admit a drawing where every edge is crossed at most k times. Since this requirement is of topological nature, the so called k-planar graphs also form a beyond-planarity graph class - in fact, they form the most studied such class (in particular for $k = 1$). Theoretical considerations of beyond-planar graph classes range from their recognition, which for almost all classes turns out to be hard, over class-inclusion relationships to so called *Turán-Type problems*, which is concerned with the maximum number of edges an n-vertex graph of this class can have.

In Chapter 7 we introduce the class of apRAC drawings which form a natural restriction of RAC graphs inspired by orthogonal drawings. We consider the inclusion relationship between apRAC and RAC graphs, provide upper bounds on the edge-density together with (almost) matching lower-bound constructions and show that the recognition problem remains NP-hard even in this more restrictive setting.

In Chapter 8, we restrict the input graphs to be bipartite and extend previous work on Turán-Type problems [10] to an additional beyond-planar graph class. In Chapter 9 we consider the class of *fan-planar* graphs in detail. We show that all three

forbidden configurations which are used to define fan-planarity are in fact necessary, as their relaxation yields a proper superclass. In the latter part of this chapter, we investigate the maximum thickness a fan-planar graph can have. The first connection between thickness and a beyond-planar graph class was established by Kainen in 1973 [124], who studied the relationship between the thickness of a graph and k-planarity. He observed that a k-planar graph G has thickness at most $k+1$. This follows from the fact that G admits by definition a drawing where every edge crosses at most k other edges, hence the edge intersection graph has maximum degree k and admits a vertex coloring using at most $k+1$ colors. Other results relating the thickness and a beyond-planar graph class C can be inferred as follows. If C has bounded edge-density, i.e., an n-vertex graph of C has at most xn many edges and if the class C is hereditary (which is true for all common classes of beyond-planar graphs), then the arboricity of C is at most x by [149] and consequently, by Eq. (3.7), the thickness of C is at most x. In particular, since n-vertex k-planar graphs have at most $3.81\sqrt{k}n$ edges [1], their thickness is at most $\mathcal{O}(\sqrt{k})$. In the case of fan-planar graphs, the edge-density argument yields an upper bound of five, which we improve to three in Chapter 9.

Experimental results In the third part of the thesis, we take a step back from the theoretical results and consider the problem of finding aesthetically pleasing drawings from an algorithmic point of view. To this end, we present in Chapter 10 an algorithmic framework that simultaneously optimizes several aesthetic criteria such as the crossing number, the angular resolution or the *stress*-value of a drawing. The benefits of our framework over previous approaches is its ease of extension to additional aesthetic criteria. In each step, our algorithm changes the drawing locally (by moving a single vertex) to guarantee efficient reevaluations of the optimization function. If the considered criterion is of local nature, we identify a set of critical vertices which are subject to change in the following iterations. To also optimize global criteria, we alternate between specific and random choices for the next vertex. Finally, in contrast to previous work that optimized several criteria at once, e.g., the work of Davidson and Harel [65], we do not use (empirically found) constants to regulate the contribution of individual criteria to the global cost function, but rather use a normalization of the values followed by an adaptive weighting scheme [112] to guarantee that the contribution of the criteria is as uniform as possible. By construction, our algorithm is a perfect candidate to postprocess already "good" drawings, e.g., ones obtained by stress minimization algorithms [99, 125].

Structure of the thesis Chapter 2 and Chapter 3 contain definitions and preliminary results used throughout the thesis. The first part contains **algorithmic results** which extend important topics of planar graphs to non-planar ones. In the second part, we obtain **combinatorial results** for a broad set of beyond-planar graph classes. In the last part, we introduce an algorithmic framework which jointly optimizes several aesthetic criteria. Finally, we summarize our results and provide concluding remarks in Chapter 11. Chapter 4 to Chapter 10 are structured as follows. First, we give a small introduction to the considered problem, which contains a related work part and ends with a paragraph that summarizes our contribution. This paragraph also contains references to the publications that the chapter may be based on. At the end of every chapter, we state open problems that arise by our work.

Chapter 2

Definitions

Most of the required definitions are stated in this chapter. Additional background in computational geometry, mathematical logic and topology can be found in [31], in [62] and in [146], respectively.

2.1 Graph Theory

Basic notation A *graph* $G = (V, E)$ is a tuple that consists of two (disjoint) sets V and E with $E \subseteq V \times V$. The elements of V are called *vertices*, while the elements of E are called *edges*. Throughout the thesis, the reader may assume that V and E are both nonempty[1]. We refer to the vertices of a graph as $V(G)$ and to the edges of a graph as $E(G)$ (independent of the actual names of these sets), which will be useful once several (auxiliary) graphs are considered simultaneously. If it leads to no confusion, we also talk about vertices (edges) of G instead of vertices (edges) of $V(G)$ (of $E(G)$). Throughout the thesis, when numbering any sort of object, we will always assume that all arithmetic on indices is modulo the size of the considered set of objects, which is usually self-evident.

Size and density The number of vertices (edges) of a graph G is denoted by $|V(G)|$ ($|E(G)|$). If it leads to no confusion, we will also abbreviate $|V(G)|$ with n and $|E(G)|$ with m throughout the thesis. The *density* of a graph is simply the quotient $\frac{|E(G)|}{|V(G)|}$. If $\frac{|E(G)|}{|V(G)|} \in \mathcal{O}(1)$, we call the graph *sparse*, otherwise we call the graph *dense*. The terms *denser* and *sparser* simply refer to graphs with higher and smaller density compared to some baseline.

Edge-vertex interaction A vertex v is *incident* to an edge e if $v \in e$ - conversely, we refer to the vertices u and v that are incident to e as the *endpoints* of e. If the graph is *undirected*, then an edge of G is an unordered tuple $\{u, v\}$ which will be abbreviated with uv in the following. Otherwise, if the direction of an edge is important, the edge $e = (u, v)$ is an ordered tuple with *source* u and *target* v. Two edges that share (at least) one endpoint are called *adjacent*, otherwise they are called *independent*. Similarly, two vertices connected by an edge are called adjacent or *neighbors*. The *neighborhood* $N(v)$ of a vertex v is the set of neighbors of v. With a slight abuse of notation, we denote by $E(v)$ the set of edges that is incident to vertex v. A graph is called *simple* if it contains neither a *self-loop*, i.e., an edge where both endpoints coincide, nor multiple edges which connect the same pair of vertices.

[1] In fact, for the covered topics that are concerned with non-planar graphs the interesting cases only arise when both $|V|$ and $|E|$ are larger than four

Degree The *degree* of a vertex v of graph G is $|E(v)|$, i.e., the number of incident edges to v. A graph G is called *degree-k* if the maximum degree of a vertex of G is k - if all vertices have degree exactly k then G is called *k-regular* - in particular, a 3-regular graph is also called a *cubic* graph.

Paths and cycles A *path* $(v_1, v_2, \ldots, v_k)$ is a sequence of vertices such that v_i and v_{i+1} are adjacent for $1 \leq i < k$. The length of a path is the number of vertices in the sequence. If additionally v_k and v_1 are also adjacent, the sequence is called a *cycle*. A graph is called *connected* if, for any pair of vertices v and u, there exists a path P such that v is the first and u is the last vertex of P, otherwise the graph is called *disconnected*. A connected graph that does not contain any cycle is called a *tree*.

Subgraph Let $G = (V, E)$ be a graph. Graph G' is called a *subgraph* of G, denoted by $G' \subseteq G$, if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$ holds. If $E(G')$ contains all edges $uv \in E(G)$ with $u \in V(G')$ and $v \in V(G')$, then G' is called an *induced subgraph*. We denote by $G[S]$ the induced subgraph of the vertices of $S \subseteq V(G)$. A (sub-)graph G is called *edge-maximal* with respect to a specific property x if G satisfies x, but the addition of any edge to G yields a graph that does not satisfy x. Similarly, we define *vertex-maximality* of a subgraph, i.e., for a subgraph $G' \subseteq G$ we have that the addition of any vertex of $V(G) \setminus V(G')$ to G' yields a graph that does not satisfy the considered property. The induced subgraphs of a graph that are vertex-maximal with respect to connectivity are called *components* of G. A property P of a graph G is called *hereditary*, if P holds for G and in addition for every induced subgraph of G.

Subdivision and contraction The *subdivision* of an edge uv removes uv from the graph, introduces a new vertex v' and connects u and v to v' - more formally, the graph G' after the subdivision of e is such that $V(G') = V(G) \cup \{v'\}$ and $E(G') = E(G) \setminus \{e\} \cup \{uv', vv'\}$. Conversely, the *contraction* of an edge uv removes uv from the graph and merges the two vertices u and v into a new vertex w. The resulting graph G' is such that $V(G') = V(G) \setminus \{u, v\} \cup \{w\}$ and $E(G') = E(G) \setminus (E(u) \cup E(v)) \cup \{wx \quad | \quad x \in (N(u) \cup N(v)) \setminus uv)\}$. Graph G is a *minor* of graph G_0 if there exists a sequence of edge contractions of G_0 which yields the intermediate graphs $G_1, G_2, \ldots$ such that $G \subseteq G_i$.

Connectivity A graph G is *k-connected*, for some $k > 0$, if there exists no set S with $|S| \leq k - 1$ vertices such that $G \setminus S$ disconnects G [2]. A 2-connected (3-connected) graph is called *biconnected* (*triconnected*). If the removal of set S disconnects G and $|S| = 1$, i.e., S contains exactly one vertex, then this vertex is called a *cut vertex*. If $|S| = 2$, then the two vertices contained in S are called a *separation pair*.

Coloring A *vertex coloring* of G with k colors is a function $c : V(G) \to \{1, 2, \ldots, k\}$ such that for any edge $uv \in E(G)$ it holds that $c(u) \neq c(v)$. Similarly, an *edge coloring* assigns a color to each edge such that no two adjacent edges have the same color. Unless otherwise specified, we assume that a coloring of graph G always refers to a vertex coloring. If we denote by $S_1, S_2, \ldots S_k$ the set of vertices obtained from the coloring, we observe that $G[S_i]$ does not contain any edge by definition - we refer to such a set as an *independent set*.

[2] It follows that 1-connected coincides with the definition of *connected*

Bipartite graphs A graph that admits a two-coloring is called a *bipartite* graph[3]. Let $G = (V, E)$ be a bipartite graph. By definition, we have that $V = A \cup B$ and $A \cap B = \emptyset$, where A and B are called the *parts* of the graph. When considering bipartite graphs in detail, we will generally denote vertices of A as a_i and vertices of B as b_i.

Clique A clique $S \subseteq V(G)$ is a set of vertices such that for any $u, v \in S, u \neq v$ we have that $uv \in E(G)$. We observe that a clique is the complement of an independent set. If $S = V(G)$, i.e., any two vertices of the given graph are adjacent, then G is called a *complete graph*. We denote the complete graph on n vertices as K_n. The *complete bipartite graph* $K_{a,b}$ is a graph with $V(G) = A \cup B$, $|A| = a, |B| = b$ and $ab \in E(G)$ for any $a \in A$ and any $b \in B$.

Decompositions A *tree decomposition* $\mathcal{T}$ of G is an auxiliary graph such that

1. $\mathcal{T}$ is a tree
2. The vertices of $\mathcal{T}$ are subsets of $V(G)$ called *bags*
3. The endpoints of every edge of $E(G)$ are both contained in at least one bag
4. If bag b_i and bag b_j contain a vertex $v \in V(G)$, then there exists a path $P = (b_i, \ldots, b_j)$ in $\mathcal{T}$ such that every bag of P contains v.

A *path decomposition* $\mathcal{P}$ of G further restricts the auxiliary graph to be a path, not an arbitrary tree. The *treewidth* of $\mathcal{T}$ is defined as $\max_i |b_i| - 1$. The treewidth $tw(G)$ of a graph G is then defined as the minimum treewidth over all valid tree decompositions. The definition for the *pathwidth* is analogous.

Graph Isomorphism Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *isomorphic* if there exists a bijective function $\phi : V_1 \to V_2$ such that

$$uv \in E_1 \Rightarrow \phi(u)\phi(v) \in E_2$$

holds for any $uv \in E_1$.

2.2 Graph Drawing

Drawing A drawing $\Gamma(G)$ of a graph G is a continuous map of G to the Euclidean plane $\mathcal{R}^2$. To be more precise, vertices of $V(G)$ are mapped to distinct points in $\mathcal{R}^2$ and edges are mapped to Jordan arcs between the images of its two endpoints. Two Jordan arcs $\gamma_1, \gamma_2 : [0,1] \to \mathbb{R}^2$ define a *meet* if $\gamma_1(x) = \gamma_2(y)$ for some $x, y \in [0,1]$ [4]. If both x and y are not in $\{0,1\}$, the meet does not coincide with the (images of) the endpoints, then γ_1 is *crossing* γ_2 if the curves switch sides [5] at the meet, which is then called a *crossing point*. If the meet is not internal, then the corresponding edges are necessarily adjacent as the vertices are mapped to distinct points. We will

[3] Equivalently, a graph is bipartite if and only if it does not contain any odd cycle

[4] There could be a pair of intervals $I_1, I_2 \subseteq [0,1]$ such that $\gamma_1(I_1) = \gamma_2(I_2)$ holds, i.e., the Jordan arcs overlap in more than a single point - however, we will impose that this will not occur in the considered drawings

[5] to be more precise, if we consider the cyclic order of the curves around the meet, a crossing occurs if the two curves alternate

assume from now on that any internal meet is in fact a crossing and the crossing point between any pair of edges is distinct [6]. If it leads to no confusion, we do not differentiate between vertices (edges) and their respective images, i.e, points (Jordan arcs) in $\mathbb{R}^2$. We call a drawing $\Gamma(G)$ *simple* if (in $\Gamma(G)$) no adjacent edges cross, no edge crosses itself and no two edges cross more than once. We will use the term *intersection* and crossing interchangeably. The drawing $\Gamma(G')$ of an (induced) subgraph $G' \subseteq G$ is obtained by restricting the drawing of $\Gamma(G)$ to the edges of G'.

Topological equivalence A *cell* of $\Gamma(G)$ is a connected component of the complement of the set of points and Jordan arcs onto which the vertices of G are mapped. Such a cell c is uniquely defined by the ordered list of vertices and crossings points which lie on the boundary δc of c. We refer to the union of these cells as the *arrangement* of Γ. Two drawings Γ_1 and Γ_2 are called *topologically equivalent* if the two arrangements are *homemorphic*, i.e., there exists a topologically preserving isomorphism between the arrangements of Γ_1 and Γ_2. If two arrangements are topologically-equivalent, they necessarily imply the same *rotation system* of the vertices and crossings, i.e., the cyclic order of the edge-(segments) incident to the vertices and crossings is equivalent in both Γ_1 and Γ_2. The set of drawings with the same rotation system form an equivalence class; we call the corresponding class a *combinatorial arrangement*.

Geometric drawings A drawing $\Gamma(G)$ of a graph G is called a *geometric drawing* if for any edge $e = uv$ of G we have that $\gamma_e : [0,1] \to \mathbb{R}^2$ with $\gamma_e(0) = \gamma_u, \gamma_e(1) = \gamma_v$ and $\gamma_e(x) = x\gamma_e(0) + (1-x)\gamma_e(1)$, i.e, γ_e is a straight line connecting (the images of) u and v. A k-bend drawing of G is a *polyline* drawing where every edge is drawn as a chain of at most $k+1$ segments. The point in which two consecutive segments touch is then called a *bend* of the edge. The *area* of a geometric drawing is the size of the smallest rectangle that contains all vertices and edge segments in its interior, where the vertices and bends must lie on integer coordinates.

If the edges of G are mapped to arbitrary Jordan arcs, then the drawing is also called a *topological* drawing. In the remainder, we always assume that the described drawing is topological unless otherwise specified.

Intersection graph and planarization Let G be a graph and $\Gamma(G)$ a drawing of G in the plane and $C \subset E \times E$ be the set of crossings of $\Gamma(G)$. Then, the *intersection graph* $\mathcal{I}$ of $\Gamma(G)$ is such that $V(\mathcal{I}) = E(G)$ and $E(\mathcal{I}) = C$, i.e., we have a vertex in $\mathcal{I}$ for every edge of G and two vertices of $\mathcal{I}$ are adjacent if the two corresponding edges cross in $\Gamma(G)$. A *planarization* $\mathcal{P}$ of $\Gamma(G)$ substitutes each crossing of C with a degree-4 vertex - in particular, for a crossing between e_1 and e_2 this is equivalent to subdividing e_1 with vertex v_1 and e_2 with vertex v_2 and identifying $v_1 = v_2$.

Planarity If $\Gamma(G)$ is a drawing without a crossing, then $\Gamma(G)$ is called an *embedding* of G. A graph G is called *planar* if there exists an embedding of G in $\mathbb{R}^2$. A planar embedding of G divides the plane into nonempty[7] regions, which we denote as *faces*. Observe that the faces of G form a special arrangement, as the boundary of any cell

[6] We remark that this is not a restriction - namely, for any internal meet I which is not a crossing or if I is a crossing point where more than two edges intersect, we can define an ϵ-neighborhood of I in $\mathbb{R}^2$ which contains only the involved edges and hence a slight perturbation of the edges inside this neighborhood will solve the issues

[7] here, nonempty refers to the fact that the regions are not degenerate

of the arrangement only contains vertices and edges. We call the equivalence class of $\Gamma(G)$ a *combinatorial embedding*.

The length of a face f, denoted by $|f|$, is equivalent to the number of vertices of δf. A vertex that occurs several times n δf will contribute several times to the length of f. The unbounded face of the drawing will be called the *outer face*. The set of faces of $\Gamma(G)$ will be denoted by $F(\Gamma(G))$ and from now on abbreviated as $F(G)$ if it leads to no confusion. If G is an edge-maximal planar graph, then we have $|f| = 3$ for every face f of G and G is called a *triangulation*.

Genus The genus y of a surface S is the maximum number of closed curves (which are contained inside S and whose pairwise intersection is empty) such that they do not disconnect S. For example, the genus of the plane is 0, as any closed curve divides the plane into two disjoint regions [122]. The genus $y(G)$ of a graph G is the minimum number y such that G has an embedding on a surface with genus y. For example, $y(K_4) = 0$, while $y(K_5) = 1$. A surface of genus one is called a *torus*. Unless otherwise specified we will always assume that a graph is drawn in the plane.

Partitions The *thickness* of a graph G is the minimum number of subgraphs $G_1, \dots G_k$ such that $E(G) = \bigcup_i E(G_i)$ and G_i is a planar graph for any $1 \leq i \leq k$. The *arboricity* is defined similarly, but, in addition, every subgraph G_i is required to be a *forest*, i.e., a graph where each component is a tree.

Crossing number The *crossing number* of a drawing $\Gamma(G)$ is simply the number of pairwise crossing edges. The crossing number of G is the minimum number of crossings over all drawings of G. It can be shown that a drawing with the minimum number of crossings is necessarily simple [160].

2.2.1 Beyond-planar graph classes

k-planar graphs A graph G is called k-planar if G admits a drawing $\Gamma(G)$ in the plane such that every edge is crossed at most k times. Schaefer [160] showed that, contrary to the crossing number, k-planar graphs might require non-simple drawings in order to obtain the minimum number of local crossings for $k \geq 4$.

Gap-planar graphs A graph G is called *k-gap-planar* if G admits a drawing $\Gamma(G)$ such that there exists an assignment of the induced crossings C of $\Gamma(G)$ to the edges of G that satisfies the following properties:

1. A crossing between edges e_1 and e_2 is assigned to either e_1 or e_2.
2. Any edge of G is assigned at most k crossings.

For an alternative view, consider the intersection graph $\mathcal{I}$ of drawing $\Gamma(G)$. The assignment can then be interpreted as an orientation of the edges of $\mathcal{I}$ such that any vertex of $\mathcal{I}$ has at most k incoming edges, i.e., any vertex of $\mathcal{I}$ is the target of at most k edges.

RAC graphs A graph G is a *RAC* (right-angle crossing) graph if G admits a geometric drawing $\Gamma(G)$ such that the smallest angle formed by any two crossing edges in $\Gamma(G)$ is exactly 90°.

2.3 Complexity Theory

We are concerned with *decision problems*, i.e., for any given input to the problem, the output will either be "yes" or "no". Fix such a decision problem A.

Computational complexity classes We say that A belongs to class P if there exists a deterministic algorithm that can solve A in polynomial time with respect to the input size of A. Throughout this thesis, we will consider decision problems on graphs and assume that the input size of A is the number of vertices of the graph unless otherwise specified.

A problem A belongs to class *NP* if, given a solution to a "yes"-instance of A, we can verify in polynomial time using a deterministic algorithm if the solution is correct. The class of *NP-complete* problems forms a subclass of NP. In particular, a problem A is NP-complete if A belongs to NP and every problem A' in NP can be reduced to A in polynomial time (using a deterministic algorithm). Finally, a problem A is called *NP-hard* if there exists an NP-complete problem A' such that A' can be reduced to A in polynomial time. Observe that NP-hard problems do not necessarily have to be in NP - hence, the verification of a solution to an NP-hard problem might be impossible in polynomial time. In fact, NP-hard problems are not required to be decision problems, but can be *optimization problems* instead.

Approximations Let X be an optimization problem, in particular, assume that X is a minimization problem. An algorithm A is called a *k-approximation* of X, if, for any instance x of X, we have that the solution-value $f(x)$ of the solution provided by A is at most $k \cdot OPT$, where OPT is the optimal solution of x. The class *APX* contains all optimization problems which admit k-approximation algorithms for constant k and polynomial running time. A polynomial-time approximation scheme (*PTAS*) is an algorithm, which takes an optimization problem X and a parameter $\epsilon > 0$ and yields a $(1+\epsilon)$-approximation algorithm for X whose running time is polynomial for a fixed value of ϵ.

If we assume that P $\neq$ NP, then every problem that is *APX-hard* does not admit a PTAS. Similar to NP-hard problems, APX-hard problems do not necessarily belong to APX.

Chapter 3

Preliminaries and Tools

This chapter is supposed to list some (elementary) results for completeness and to refer to later on. Some of them are taught in class, some of them are foundational results and some of them can be immediately derived from the definitions.

Graphs Since any edge has exactly two endpoints, the following relation immediately follows for any graph G, which is widely known as the *Handshaking Lemma*

$$\sum_{v \in V(G)} deg(v) = 2|E(G)| \tag{3.1}$$

Planar graphs Let G be a planar graph and $\Gamma(G)$ be a (planar) embedding of G. Denote by c the number of connected components of G and let F be the set of faces of $\Gamma(G)$. *Euler's Formula* then relates the number of vertices, edges, faces and connected components as follows.

$$|V(G)| + |F(G)| - |E(G)| = c + 1^{1} \tag{3.2}$$

The formula can be for example proven using induction on the vertices. We direct the interested reader to [85], which lists 21 different proofs of the statement. Let G be a planar graph and let $\Gamma(G)$ be an embedding of G. As any edge belongs to exactly two facial walks, we have the following relation for planar graphs:

$$\sum_{f_i \in F(G)} |f_i| = 2|E(G)| \tag{3.3}$$

Let G be a simple planar graph. Then,

$$|E(G)| \leq 3|V(G)| - 6 \tag{3.4}$$

If G is in addition bipartite, then

$$|E(G)| \leq 2|V(G)| - 4 \tag{3.5}$$

We provide a short proof here for completeness: Clearly, if G is not connected, the addition of an edge to G does not break planarity (since at least one vertex of each component is on the outer face), thus we can assume that G is connected since we aim to provide an upper bound on the number of edges. Since G is simple, it contains no self-loop nor multi-edges, hence any face has length at least three. Using Eq. 3.3, we obtain $3|F(G)| \leq 2|E(G)|$. We substitute $|F(G)|$ in Eq. 3.2 with this inequality and obtain $|V(G)| + \frac{2}{3}|E(G)| - |E(G)| \geq 2$. Solving for $|E(G)|$ yields $|E(G)| \leq$

[1] If the given graph is connected we have $c + 1 = 2$.

$3|V(G)| - 6$. For the bipartite case, we observe that a face of length k implies a cycle of length k in G – it follows that a bipartite graph cannot have odd length faces. Hence, any face has length at least four and by repeating the argument above we obtain $4|F(G)| \leq 2|E(G)|$ and $|V(G)| + \frac{1}{2}|E(G)| - |E(G)| \geq 2$ which yields the desired result.

Crossing Lemma The famous *Crossing Lemma* [3] relates the number of crossings of a graph G to its number of vertices and edges. Namely,

$$cr(G) \geq \frac{|E(G)|^3}{c \cdot |V(G)|^2} \tag{3.6}$$

We remark here that this lower bound is asymptotically tight as it can be shown that the complete graph K_n with n vertices and $\Theta(n^2)$ edges has $\Theta(n^4)$ crossings. The initial work showed the statement with the constant $c = 64$ for graphs where $|E(G)| \geq 4|V(G)|$ holds. If we restrict the lemma to only work for denser graphs, we can improve the leading constant. The current best constant of $c \approx 29$ [1] holds for graphs with $|E(G)| \geq 7|V(G)|$.

Arboricity and thickness Let G be a graph and denote by $a(G)$ and $t(G)$ its arboricity and thickness, respectively. Consider the following equation which relates the arboricity and the thickness.

$$\frac{a(G)}{3} \leq t(G) \leq a(G) \tag{3.7}$$

The upper bound for $t(G)$ holds as any forest is a planar graph. For the lower bound, we observe that any planar graph can be decomposed into three forests[2], hence it follows that $t(G) \geq \frac{a(G)}{3}$. Computing the thickness of a given graph (or even deciding if the thickness is two or not) is in general NP-complete [145]. However, as the arboricity can be computed in polynomial time [101], there exists a 3-approximation to the thickness. By the Nash-Williams theorem [149], the arboricity of G is $a(G) = \max_S \frac{|E(S)|}{|V(S)|-1}$, where S ranges over all subgraphs of G.

Rerouting of edges A common tool to slightly alter a drawing Γ is to locally change the curve of an edge e, which will be called a *rerouting* of the edge e. Let $S = (e_1, e_2, \ldots, e_k)$ be an ordered sequence of edges and let $e_i = u_i v_i$. We say that an edge $e = (u, v)$[3] can be *drawn along* S if the following conditions hold:

1. $u \in e_1$,
2. $v \in e_k$ and
3. for any $1 \leq i \leq k-1$, the edge e_i shares a common point x_i with e_{i+1} (which is usually an intersection point).

We then draw e along S by starting at u, we follow the curve of e_1 until x_1, follow the curve of e_2 until x_2 and repeat until we reach v.

[2] such a decomposition can be for example derived from [163]

[3] the edge e may be undirected, however, when routing an edge specifically it is easier to give it a direction and draw it from a specified source to a target

Part I

Algorithmic constructions

Chapter 4

Exact and approximate k-planarity testing for graphs of small pathwidth

4.1 Introduction

We consider the following fundamental problem of graph theory in this chapter. Given a graph G and a class of graphs $\mathcal{C}$ - can we efficiently test if G is contained in $\mathcal{C}$? As discussed in the introduction, this can be answered in the affirmative if class C is minor-closed, e.g., if $\mathcal{C}$ corresponds to the class of planar graphs. Unfortunately, many extensions of planar graphs are not minor-closed and for some the recognition problem is even known to be NP-complete. We consider in the following the recognition problem for the class of k-planar graphs.

Related work The recognition of k-planarity was mainly studied for $k = 1$. It is NP-hard [108, 136] to decide if a given graph is 1-planar. The problem remains NP-hard for bounded pathwidth graphs [22]. Positive results were obtained for some restricted classes of 1-planar graphs. Namely, one can test in linear time if the graph is (i) outer-1-planar [19], (ii) optimal 1-planar [43] or (iii) maximal 1-planar if we assume that the rotation system (i.e., the circular ordering of the edges around each vertex) is given [82]. We remark that the assumption of a given rotation system does not make the result trivial, as it can be shown that the recognition problem of 1-planarity remains NP-hard for 3-connected graphs even if the rotation system is provided [20].

For general k, the work of [178] shows that the recognition of k-planarity is NP-hard for arbitrary k. Moreover, they show that the corresponding minimization problem is APX-hard (in fact, it can not be approximated by a constant smaller than $2 - \epsilon$). Regarding positive results, it can be shown [96] that simple optimal 2-planar graphs can be recognized in linear time.

The restriction to graphs with small pathwidth was used in [170] to obtain approximation algorithms for the crossing number problem of a graph G.

Our contribution We study exact and approximate k-planarity testing for graphs of small pathwidth. Besides providing some classes of graphs for which 1-planarity testing can be done efficiently, we will also provide algorithms that yield so called *x-approximations* to k-planarity. An x-approximation to k-planarity either yields a drawing in which every edge is crossed at most $(x \cdot k)$ times - or we can certify that the input graph cannot be k-planar.

In Sections 4.3 and 4.4 we provide exact algorithms which recognize (in linear time) whether 3- or 4-paths, which are maximal pathwidth-3 and pathwidth-4 graphs, respectively, are 1-planar, thus extending the class of graphs for which testing 1-planarity can be done efficiently. In Section 4.5, we consider pathwidth-3 graphs and provide an algorithm that yields a 6-approximation to k-planarity for $k \leq 4$ and a 7-approximation for general k. Finally, in Section 4.6, we consider w-paths for arbitrary w and provide a construction which yields a $\mathcal{O}(w)$-approximation to k-planarity.
This chapter is based on joint work with Miriam Münch and Ignaz Rutter titled "Exact and approximate k-planarity testing for maximal graphs of small pathwidth" which was recently accepted at the "50th International Workshop on Graph-Theoretic Concepts in Computer Science (WG2024)".

4.2 Preliminaries

We briefly recall relevant concepts from Chapter 2. A *k-planar* graph G is a graph that admits a drawing Γ such that every edge is crossed at most k times in Γ. A *path-decomposition* is a sequence of subsets of vertices of G such that the endpoints of each edge appear in one of the subsets and such that each vertex appears in a contiguous subsequence of the subsets. The *pathwidth* of such a decomposition is one less than the size of its largest set. The pathwidth $pw(G)$ of a graph G is the smallest width among all possible path-decompositions of G. A *w-path* is a graph G of pathwidth w such that in any width-w path decomposition of G each bag forms a clique. Adding any edge in a w-path increases its pathwidth, this is why they are also called *maximal pathwidth-w graphs*. In the following, we will use a special type of path decomposition introduced by Biedl et. al. [170]. A path decomposition P of G of width w is *alternating* if P contains exactly $l = 2|V(G)| - 2w - 1$ bags such that $|X_i| = w + 1$ if i is odd and $|X_i| = w$ if i is even with $X_{i-1} \supset X_i \subset X_{i+1}$ for every even $1 < i < l$; see Figure 4.1.

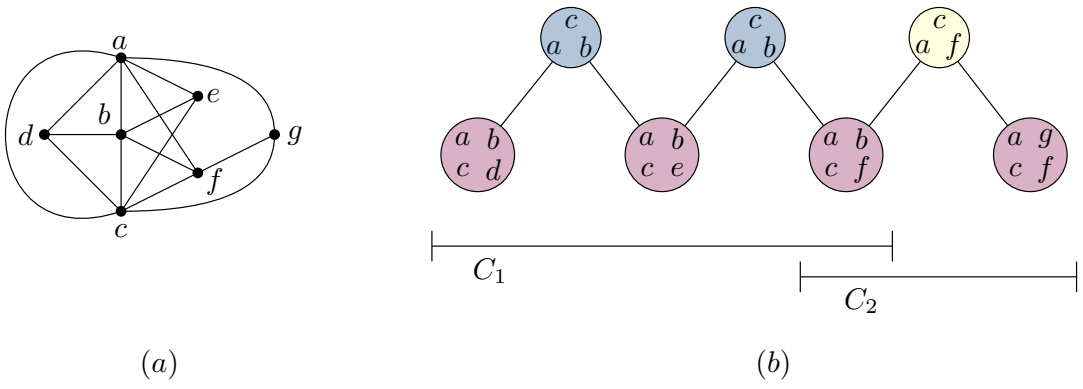


FIGURE 4.1: A 3-path G (a) with alternating path decomposition P of width 3 (b). P contains two clusters C_1, C_2 with anchor triplet $\{a, b, c\}$ and $\{a, c, f\}$, respectively.

Let G be a graph of pathwidth w and let $P = X_1, \ldots, X_l$ be an alternating path decomposition of G. For a bag X_i with $|X_i| = w$, the *cluster* C is the maximal (consecutive) set of bags which all contain X_i; i.e. $C = \{X_p \mid X_i \subseteq X_p\}$; see Figure 4.1. By definition, any bag X_i is contained in at most two clusters - if X_i is contained in exactly one cluster C with anchor set Y, then we will also call Y the anchor set of X_i. If X_i is contained in two clusters C_a and C_b with $a < b$, then the anchor set of X_i will

be the anchor set of C_a (in particular, this anchor set is then X_{i-1}). In the special case of $w = 3$, we call the anchor sets *anchor triplets* and denote them by T.

Lemma 1 ([170]). *Let G be a graph with a path decomposition $P = X_1, \ldots, X_l$ of width w and let $v \in X_1$. Then G has an alternating path decomposition $P' = X'_1, \ldots, X'_k$ of width w such that contains $v \in X'_1$. It can be found in $O(w \cdot |P|)$ time.*

Let X be an arbitrary bag of size $w+1$ in an alternating path decomposition of width w of a w-path G such that Y is the anchor set of X. If vertex $d \in V(G)$ is connected to $1 \leq k \leq w$ vertices of Y, then d is called a *k-stack* of the set $N(d) \cap Y$. In Figure 4.1, vertex f is a 3-stack of $\{a, b, c\}$. We will call a k-stack *proper*, if it does not belong to any anchor set - in our example, vertex f is not a proper 3-stack as it is contained in the anchor set of C_2, but vertex e is. Let Γ be a 1-planar drawing of a 1-planar graph G with pathwidth w and let $X_1, \ldots, X_l$ be the bags of the path decomposition. We denote by $\Gamma[i]$ the subdrawing of Γ induced by the vertices of X_i. Moreover, we denote by $\Gamma[i:j]$ the subdrawing of Γ restricted to the vertices of $\bigcup_{s=i}^{j} X_s$. We write $\Gamma[:j]$ for $\Gamma[1:j]$ and $\Gamma[i:]$ for $\Gamma[i:l]$. Similarly we denote by $G[i:j]$ the subgraph of G induced by the vertices of $\bigcup_{s=i}^{j} X_s$ and write $G[:j]$ for $G[1:j]$ and $G[i:]$ for $G[i:l]$.

In order to guarantee that a given graph cannot be k-planar, we will leverage the following tools.

Lemma 2. *$K_{3,7k+1}$ is not k-planar.*

Proof. The crossing number of $K_{3,a}$ is $\lfloor \frac{a}{2} \rfloor \lfloor \frac{a-1}{2} \rfloor$ [156]. Since $K_{3,a}$ has $3a$ edges

$$lcr(K_{3,a}) \geq \frac{2}{3a} \cdot \lfloor \frac{a}{2} \rfloor \lfloor \frac{a-1}{2} \rfloor = \begin{cases} \frac{2}{3a} \cdot \frac{a}{2} \cdot \frac{a-2}{2} = \frac{a-1}{6}, & \text{if } a \text{ even} \\ \frac{2}{3a} \cdot \frac{a-1}{2} \cdot \frac{a-1}{2} = \frac{(a-1)^2}{6a}, & \text{otherwise} \end{cases}$$

Choosing $a = 7k+1$ then yields the desired result. □

In fact, for small k, we can obtain stronger results.

Lemma 3 ([135],[11]). *$K_{3,3+4k}$ is not k-planar for $k \leq 4$.*

Lemma 4. *A w-path is not k-planar if $\frac{w+2}{7.62} > \sqrt{k}$.*

Proof. A w-path G contains an induced K_{w+1} by definition, which is a subgraph on $w+1$ vertices with $\frac{(w+1)(w+2)}{2}$ edges. Since an n-vertex k-planar graph has at most $3.81\sqrt{k}n$ edges [1], G cannot be k-planar if $\frac{w+2}{2} > 3.81\sqrt{k}$ holds. □

For the special case of 1-planarity, we have stronger tools at hand.

Lemma 5 ([135]). *The complete graph K_7 with three missing edges which form a K_3, denoted by $K_7^{-\triangle}$, is not 1-planar.*

The following graph on 8 vertices is also not 1-planar and might be of interest independently. Let C_5^+ denote a 5-cycle with a chord and let K_6^- be the graph we obtain from the complete graph on 6 vertices by deleting one edge. In the remainder, let $H = K_8 - C_5^+$, see Figure 4.2 for an illustration of H and its complement.

By construction, H consists of vertices v_1,v_2 and v_3 with degree seven, vertices v_4,v_5 and v_7 of degree five and vertices v_6 and v_8 of degree four. Consider now the subgraphs $H_1 = \{v_1, v_2, v_3, v_4, v_6, v_7\}$ and $H_2 = \{v_1, v_2, v_3, v_4, v_5, v_7\}$ and observe the following properties:

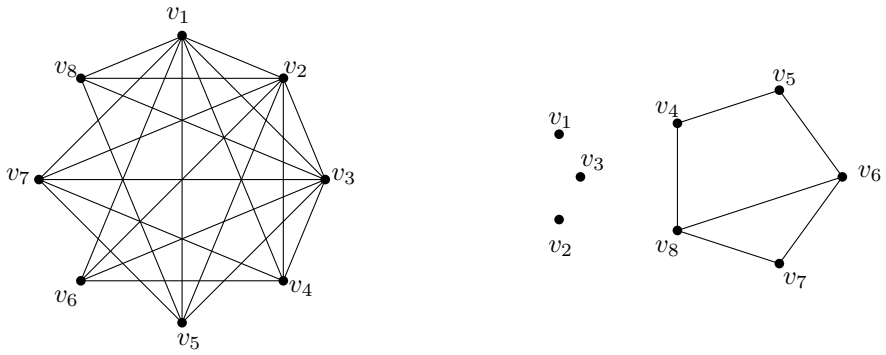


FIGURE 4.2: Graph H and its complement.

- H_1 and H_2 are both isomorphic to K_6^-
- v_1, v_2 and v_3 are contained in both H_1 and H_2
- $v_8 \notin H_1 \cup H_2$
- $v_5v_8 \in H$
- $v_5 \in H_2$ and $v_5 \notin H_1$.

Lemma 6. *The graph $H = K_8 - C_5^+$ is not 1-planar.*

Proof. Since $H_1 = K_6^-$ it has exactly three 1-planar embeddings up to isomorphism [135]; see Figure 4.3.

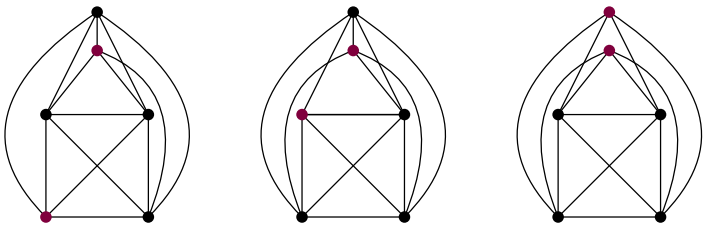


FIGURE 4.3: The three non-isomorphic 1-planar drawings of K_6^-. The edge between the red vertices is missing.

To extend these embeddings to an embedding of H consider the addition of vertex v_5. By construction, v_5 has to be connected to three black and exactly one red vertex of Figure 4.3. One can easily see that this is impossible for the middle and the right drawing, while we obtain six non-isomorphic 1-planar drawings after inserting v_5 in to the left drawing of H_1, see Fig. 4.4.

Finally, to complete our drawing to H, we have to add vertex v_8 - which is connected to v_5, i.e., the blue vertex and in addition to the three black neighbors of v_5. However, as shown in Fig. 4.5 we have in every case that a subset of the crossed edges (fat in Figure) bound a region that contains neighbors of v_8 both in the interior and the exterior - and since we cannot intersect the boundary of this region, we obtain the desired result.

□

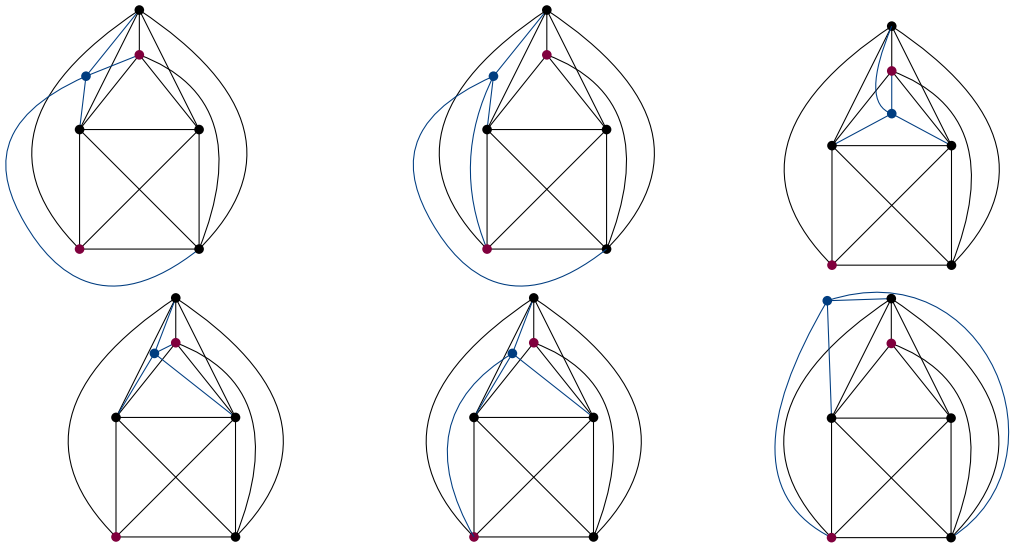


FIGURE 4.4: Six non-isomorphic 1-planar drawings of $H_1 + v_7$. Vertex v_7 is drawn in blue.

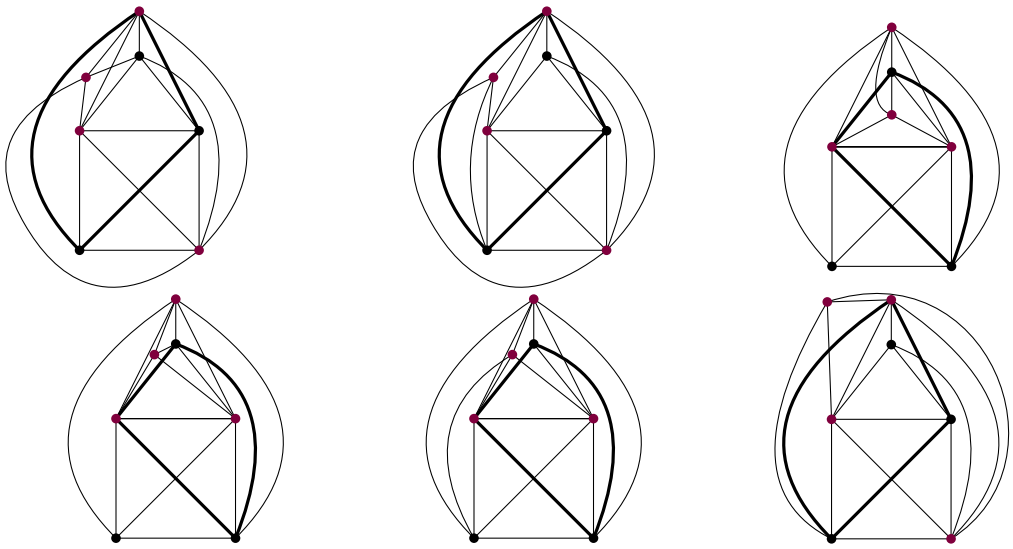


FIGURE 4.5: The bold edges cannot be crossed anymore and form a region that contains neighbors of v_8 (red vertices) both in the interior and exterior and thus no 1-planar drawing of H is possible.

4.3 3-Paths

In this section, we will provide an efficient algorithm to test if a 3-path is 1-planar. For the remainder of this section let G be a 1-planar 3-path and let $P = X_1, \ldots, X_l$ be an alternating path decomposition of G. Let $C_1, \ldots, C_s$ be the corresponding clusters with corresponding anchor triplets $T_1, \ldots, T_s$ in the order given by P. Further, let X be the set of vertices of G that are contained in at least one anchor triplet. We call the subgraph $G[X]$ of G induced by X the *skeleton* of G.

Our first goal is to show that if G is 1-planar, then there exists a 1-planar drawing Γ such that $\Gamma[X]$ is planar. To this end, we will first establish a structural property of low degree vertices.

Lemma 7. *Let z be a vertex of degree 3 in G such that the subgraph H induced by $z \cup N(z)$ is isomorphic to K_4. If $\Gamma[H]$ is not planar, then there exists a 1-planar drawing Γ' of G s.t.*

(i) $\Gamma'[H]$ is planar,

(ii) any two crossing edges of Γ' cross also in Γ, and

(iii) all vertices and edges of G that are not in H are drawn in the face of $\Gamma'[H]$ that is not incident to z.

Proof. Denote the vertices of H by u, z, v, w such that uz crosses vw in Γ by assumption. By 1-planarity, the cycle $uvzw$ can be drawn without crossings in such a way that its interior contains only the two crossing edges uz and vw. Then $\Gamma - z$ contains an empty triangular face that is incident to u, v, w and by inserting z into this face, we obtain a 1-planar drawing Γ' of G that satisfies all three conditions. □

We will now leverage this lemma to prove the following statement.

Lemma 8. *There exists a 1-planar drawing Γ of G such that $\Gamma[i]$ is planar for each $1 \leq i \leq l$.*

Proof. Let Γ be a 1-planar drawing of G that minimizes the number of crossings. Assume for the sake of contradiction that there is a bag i such that $\Gamma[i]$ is non-planar. If there is a vertex z in X_i such that X_i is the only bag that contains z, then $\deg_G(z) = 3$ and the vertices of X_i induce a K_4 by maximality. Hence, we can apply Lemma 7 to obtain a 1-planar drawing with fewer crossings; a contradiction.

We may hence assume that $1 < i < l$ and that $X_i = \{u, v, w, z\}$ with $z \notin X_{i-1}$ and $u \notin X_{i+1}$. Observe that z has degree 3 in $G[1 : i]$ and by applying Lemma 7 to $\Gamma[1 : i]$, we obtain a drawing Γ_1 of $G[1 : i]$ with fewer crossings than $\Gamma[1 : i]$ such that $\Gamma_1[i]$ is planar and all other vertices and edges are drawn inside the face of $\Gamma_1[i]$ incident to u, v, w. Likewise, u has degree 3 in $G[i : l]$ and by applying Lemma 7 to $\Gamma[i : l]$, we obtain a drawing Γ_2 of $G[i : l]$ with fewer crossings than $\Gamma_2[i : l]$ such that $\Gamma_2[i]$ is planar and all other vertices and edges are drawn inside the face of $\Gamma_2[i]$ incident to v, w, z. Combining the drawings Γ_1 and Γ_2 yields a 1-planar drawing of G with fewer crossings than Γ; a contradiction. □

This observation allows us to transform an arbitrary 1-planar drawing Γ of G into a 1-planar drawing Γ' of G such that $\Gamma'[i]$ is planar for every $1 \leq i \leq l$ and any two crossing edges of Γ' cross also in Γ.

Hence, we can assume from now on that $\Gamma[i]$ satisfies Lemma 8 for any $1 \leq i \leq k$. In the next lemma, we will extend this result to $\Gamma[X]$.

Lemma 9. *There exists a 1-planar drawing Γ of G such that the subdrawing $\Gamma[X]$ is planar.*

Proof. Let Γ be a 1-planar drawing of G. By Lemma 8 we may assume that each bag of P induces a plane drawing in Γ. Assume that edges of two distinct anchor triplets $T_i = \{a_i, b_i, c_i\}$ and $T_j = \{a_j, b_j, c_j\}$ cross in Γ. Without loss of generality assume $i < j$ and that there is no index $i < k < j$ such that T_k crosses T_i or T_j. Note that $i \neq j - 1$, since then P would contain a bag $X = T_i \cup T_j$, a contradiction to Lemma 8. Further observe that for every anchor triplet T_k in G there exists a vertex $x_k \in G[: k]$ with $T_k \subseteq N(x_k)$ and there is a vertex $y_k \in G[k :]$ with $T_k \subseteq N(y)$ and $x_k \neq y_k$. Note that $x_i \notin T_j$ and $y_j \notin T_i$. We distinguish cases based on the number of vertices shared by T_i and T_j.

First consider the case that T_i and T_j are disjoint. Up to a renaming of the vertices, by 1-planarity $\Gamma[T_i \cup T_j]$ has to be embedded as in Fig. 4.6(a). Since both triangles T_i and T_j have only one uncrossed edge, there is only one possible way to place x_i

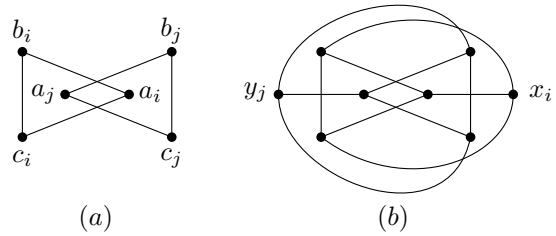


FIGURE 4.6: (a) Two vertex-disjoint crossing anchor triplets. (b) The unique 1-planar embedding of the graph induced by T_i, T_j, x_i and y_j.

and y_j, respectively; see Fig. 4.6(b). Then, no further 3-stack to T_i or to T_j can be added without loosing 1-planarity, hence $y_i \in T_j$, $x_j \in T_i$. The only vertex of T_j from which all vertices in T_i can be reached while maintaining 1-planarity is a_j. Since by assumption there is no anchor triplet between T_i and T_j that crosses T_i or T_j, we have $T_{i+1} = \{a_j, b_i, c_i\}$. No vertex adjacent to all vertices in T_{i+1} can be added without loosing 1-planarity, hence $y_{i+1} \in T_j$. But neither from b_j nor from c_j all vertices in T_{i+1} can be reached. Hence we may assume that $T_i \cap T_j \neq \emptyset$.

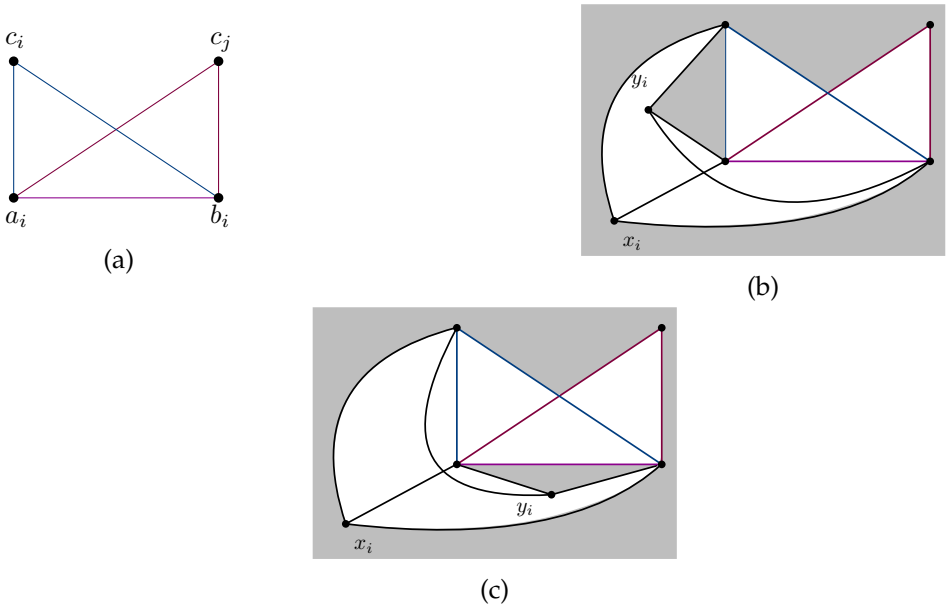


FIGURE 4.7: (a) Two crossing anchor triplets that share the purple edge. $(b), (c)$ The two non-isomorphic embeddings of the subgraph induced by T_i, T_j, x_i and y_i.

Next we consider the case $|T_i \cap T_j| = 2$. Without loss of generality assume that $a_i = a_j, b_i = b_j, c_i \neq c_j$; see Fig. 4.7$(a)$. First observe that $y_i \in T_j$ implies $y_i = c_j$ and hence $T_{i+1} = \{a_i, b_i, c_j\} = T_j$, which contradicts $i \leq j - 1$. Hence $y_i \notin T_j$. Next, observe that there are only two 1-planar embeddings of $G[T_i \cup T_j \cup \{x_i, y_i\}]$; see Fig. 4.7$(b)$, (c). Since y_j has degree 3 it cannot be placed inside a triangular face bounded by a crossing and a vertex not in T_j. Hence only the shaded faces remain, but from none of them all vertices in T_j can be reached while maintaining 1-planarity and we obtain a contradiction.

It remains to consider the case $|T_i \cap T_j| = 1$. Without loss of generality, assume that $a_i = a_j$ and b_i, b_j, c_i, c_j are pairwise distinct. Then there are two subcases – either (i) exactly one of the crossing edges is incident to a_i; see Fig. 4.8(a), or (ii) none of them is incident to a_i; see Fig. 4.8(b).

FIGURE 4.8: (a) Two crossing anchor triplets, where exactly one of the crossing edges is incident to the shared vertex. (b) Two crossing anchor triplets, where none of the crossing edges is incident to the shared vertex.

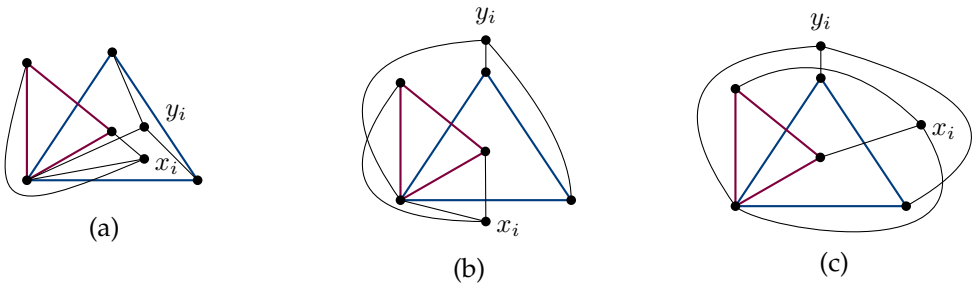


FIGURE 4.9: The non-isomorphic embeddings of $G[T_i \cup T_j \cup \{x_i, y_j\}]$ in Case (i).

Case (i): While x_i can be placed in each of the four faces of the drawing in Fig. 4.8(a), only three of them admit a 1-planar drawing when placing y_j - moreover, each of them implies a unique routing of the edges as well as a unique placement for y_j and its incident edges; see Fig. 4.9. In neither of these drawings it is possible to add another vertex that is adjacent to all vertices in T_i (red triangle). Thus we conclude $y_i \in T_j \setminus \{a_j\}$. Similarly we get $x_j \in T_i \setminus \{a_i\}$. This implies that y_i and x_j are contained in every anchor triplet T_k with $i < k < j$. Since also $a_i = a_j$ is contained in every T_k with $i < k < j$ it follows that $i = j - 2$ and $T_{i+1} = \{a_i, y_i, x_j\}$. In particular we have $y_{i+1} = T_j \setminus \{a_j, y_i\}$.

We now consider the embeddings from Fig. 4.9 separately. In Fig. 4.9(a) only b_j can reach all vertices of T_i while maintaining 1-planarity, thus $y_i = b_j$. For either choice of $x_j \in \{b_i, c_i\}$ we can connect c_j to all vertices in T_{i+1} without loosing 1-planarity. Hence assume $x_j = b_i$ without loss of generality. For no $S_i \in \binom{T_i}{2}$ it is possible to insert a vertex adjacent to all vertices in $S_i \in \binom{T_i}{2} \cup \{x_i\}$ while maintaining 1-planarity. Similarly, for no $S_j \in \binom{T_j}{2}$ it is possible to insert a vertex adjacent to all vertices in $S_j \in \binom{T_j}{2} \cup \{y_j\}$. Thus G is induced by T_i, T_j, x_i and y_j and thus can be drawn without introducing any crossings; see Fig. 4.11. The remaining cases as shown in Fig. 4.9$(b) - (c)$ work analogously; i.e. we get that G is the graph induced by T_i, T_j, x_i and y_j for every possible choice of y_i and x_j.

Case (ii): In this case, we only have two possibilities to place x_i while maintaining 1-planarity, see Fig. 4.10. Drawing x_i only leaves one possible position for y_j. In none of the two drawings it is possible to insert a new vertex adjacent to all vertices in T_i, thus $y_i \in T_j \setminus \{a_j\}$. Similarly we get $x_j \in T_i \setminus \{a_i\}$. This again implies that y_i and x_j are contained in every anchor triplet T_k with $i < k < j$. Since also $a_i = a_j$ is contained in every T_k it follows that $i = j - 2$ and $T_{i+1} = \{a_i, y_i, x_j\}$. In particular,

(a) (b)

FIGURE 4.10: The non-isomorphic embeddings of $G[T_i \cup T_j \cup \{x_i, y_j\}]$ in Case (ii).

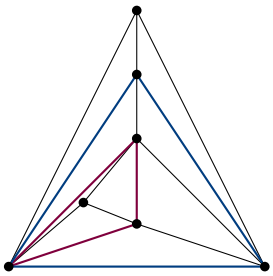


FIGURE 4.11: A planar drawing of $G[T_i \cup T_j \cup \{x_i, y_j\}]$.

we have $y_{i+1} = T_j \setminus \{a_j, y_i\}$.

In Fig. 4.10(a), we can reach all vertices of T_i from both b_j and c_j, hence either $y_i = b_j$ or $y_i = c_j$. W.l.o.g. we assume that $y_i = b_j$, the other case is symmetric. We observe that for either choice of $x_j \in \{b_i, c_i\}$ we can connect c_j to all vertices in T_{i+1} without loosing 1-planarity. Hence assume $x_j = b_i$ without loss of generality. For no $S_i \in \binom{T_i}{2}$ it is possible to insert a vertex adjacent to all vertices in $S_i \in \binom{T_i}{2} \cup \{x_i\}$ while maintaining 1-planarity. Similarly, for no $S_j \in \binom{T_j}{2}$ it is possible to insert a vertex adjacent to all vertices in $S_j \in \binom{T_j}{2} \cup \{y_j\}$. Thus G is induced by T_i, T_j, x_i and y_j and thus can be drawn planarly; see Fig. 4.11. The remaining case shown in Fig. 4.10(b) works analogously; i.e. we get that G is the graph induced by T_i, T_j, x_i and y_j for every possible choice of y_i end x_j. □

Let Γ be a drawing of G whose restriction to the skeleton is planar and where every bag is drawn planar, which exists by Lemma 9. Since the skeleton of G is triconnected, it has a unique planar embedding. Hence we can transform Γ such that $\Gamma[X]$ remains planar and for every $1 \leq i < s$ the triangle $\Gamma[T_{i+1}]$ is drawn in the interior (or on the boundary) of $\Gamma[T_i]$. We call a drawing with these properties *nice*. For every $1 \leq i \leq s$ let i' be the smallest (even) index such that $X_{i'}$ coincides with T_i. A proper 3-stack v of an anchor triplet T_i is *inside* T_i if it is inside the triangular region defined by T_i in $\Gamma[i']$ and it is *outside* T_i if it is outside T_i in $\Gamma[i']$. Note that the fact that $\Gamma[i']$ is planar guarantees that every 3-stack of T_i lies completely inside T_i or completely outside of T_i.

Lemma 10. *Let T_i and T_{i+1} be two consecutive anchor triplets of P and let Γ be a nice 1-planar drawing of G which minimizes the number of crossing among all such drawings. Then there is either no proper 3-stack of T_i inside T_i or no proper 3-stack of T_{i+1} outside T_{i+1}.*

Proof. Assume for the sake of contradiction that u is a 3-stack inside T_i and v is a 3-stack outside T_{i+1}. We distinguish cases based on whether u is drawn inside T_{i+1}

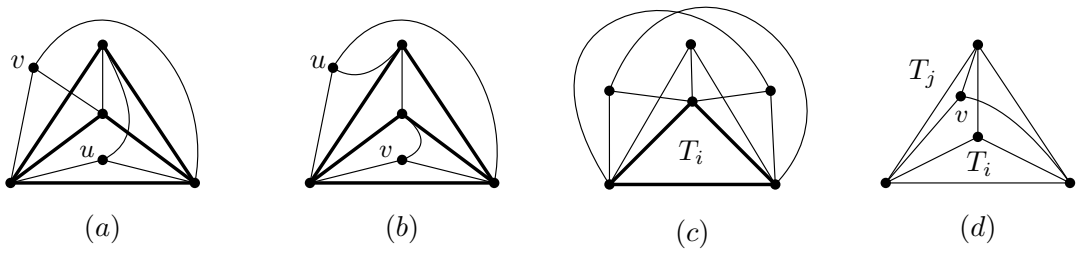


FIGURE 4.12: Illustrations for the proofs of Lemmas 10–12.

or not. If u is drawn inside T_{i+1}, then v must be drawn outside T_i; see Fig. 4.12(a). We can then swap the positions of u and v (and reroute some edges as indicated in Fig. 4.12(b)) to remove some crossings while maintaining 1-planarity; a contradiction to our choice of Γ. Hence u must be drawn in the region between T_i and T_{i+1} in Γ. A symmetric argument applies to v. However, there is no way to 1-planarly draw u and v in this region simultaneously. □

The following lemma will further restrict the number of proper 3-stacks.

Lemma 11. *Let Γ be a nice drawing of a 1-planar 3-path G. Then*

(i) no anchor triplet T_i with $i \geq 2$ has two proper 3-stacks outside T_i and

(ii) no anchor triplet T_j with $j < s$ has two proper 3-stacks inside T_j.

Proof. We show only (i) as (ii) follows from (i) by changing the outer face and reversing the path decomposition. Assume for the sake of contradiction that there is an anchor triplet T_i, $i \geq 2$ with two proper 3-stacks u, v outside T_i. Then u, v and the vertex $x \in T_{i-1} \setminus T_i$ are all 3-stacks of T_i that lie outside T_i, and they are hence drawn as shown in Fig. 4.12(c). However, since T_{i-1} is an anchor triplet, there needs to be an additional 3-stack on T_{i-1}, which is, however, impossible to add as the vertex x is already enclosed by a cycle of crossing edges. □

Let v be a proper 3-stack. We say that v is *between* T_i and T_{i+1} if T_{i+1} is drawn inside T_i and v is either a 3-stack inside T_i or a 3-stack outside T_{i+1}. Lemma 10–11 then together imply that there is at most one proper 3-stack between any two consecutive anchor triplets.

Lemma 12. *There exists a 1-planar drawing Γ of G such that (i) no two edges of $G[X]$ cross each other and (ii) every edge (a, b) where a and b belong to the same anchor triplet T is crossing-free.*

Proof. By Lemma 9 there exists a nice 1-planar drawing of G which satisfies (i) and such that every bag is drawn planar. Let Γ be the drawing that minimizes the number of crossings among all such drawings. We will now modify Γ in order to establish (ii). We first remove all proper 3-stacks from Γ that are between two consecutive anchor triplets. Hence, the only proper 3-stacks left are outside T_1 or inside T_l (or vice-versa) and hence (ii) is satisfied. We then reinsert the removed proper 3-stacks and put each 3-stack v between T_i and T_{i+1} into the region that is bounded by T_i and T_{i+1}; see Fig. 4.12(d). For a single proper 3-stack, this preserves 1-planarity. Since there is at most one proper 3-stack between any two consecutive anchor triplets and since the regions between consecutive anchor triplets in the drawing are pairwise disjoint, we can add all removed proper 3-stacks simultaneously without violating 1-planarity. □

Theorem 1. *There is a linear-time algorithm for testing whether a given 3-path is 1-planar. In the affirmative case, the algorithm yields a 1-planar drawing.*

Proof. If G is 1-planar, there exists a 1-planar drawing satisfying the conditions guaranteed by Lemma 12. We associate such a drawing with an s-dimensional vector $(x_i)_{i=1}^s$ where x_i is the number of proper 3-stacks of T_i that are outside T_i. The properties from Lemma 12 give that $x_i \in \{0,1\}$ for $i \geq 2$. Moreover, clearly $x_1 \leq 3$. Given two drawings Γ and Γ' with these properties and associated vectors $x = (x_i)_{i=1}^s$ and $x' = (x'_i)_{i=1}^s$, if x is lexicographically larger than x', let j be the first position such that $x_j > x'_j$. If $j = 1$, we can simply move an additional proper 3-stack of T_1 outside of T_1 in Γ' to obtain a drawing whose vector is lexicographically larger. We therefore may assume $j > 1$. Then $x'_j = 0$ and $x_j = 1$. Moreover, x_{j-1} equals the number p_{j-1} of proper 3-stacks of T_{j-1} as by Lemma 10 and 11 there can be at most one proper 3-stack between T_j and T_{j-1}. By the choice of j, we find $x'_{j-1} = x_{j-1} = p_{j-1}$ and hence no proper 3-stack of T_{j-1} is between T_{j-1} and T_j in Γ'. We can hence take a proper 3-stack of T_j and move it between T_j and T_{j-1}. This gives a 1-planar drawing with a lexicographically larger vector.

This implies that a drawing satisfying the conditions from Lemma 9 that has the lexicographically largest vector can be found by a simple greedy algorithm that starts with the unique planar drawing Γ' of the skeleton with T_1 on the outer face and then iteratively adds the proper 3-stacks of T_i for $i = 1, \ldots, s$. Everytime, we embed as many proper 3-stacks outside T_i as possible without violating 1-planarity and the remaining ones between T_i and T_{i+1}. If this process fails to produce a 1-planar drawing, the graph is not 1-planar. □

4.4 4-Paths

For the remainder of this section let G be a 4-path with path decomposition $P = X_1, \ldots, X_l$. We begin by observing the following properties which have to be fulfilled in order for G to be 1-planar.

Property 1. *Let Y be an anchor set of G. Then there are at most two 4-stacks on Y.*

Proof. Suppose for a contradiction that there exists a 1-planar 4-path where anchor set $Y = \{a, b, c, d\}$ is four-stacked three times with vertices $\{v_1, v_2, v_3\}$. As every bag induces a K_5, we have that the subgraph induced by these seven vertices is complete minus the set of edges defined by any pair of $\{v_1, v_2, v_3\}$ and thus the subgraph is exactly $K_7^{-\triangle}$ which is not 1-planar by Lemma 5. □

Property 2. *An anchor set Y_i can only be four-stacked twice if $i = 1$.*

Proof. Suppose for a contradiction that anchor set $Y_i = \{a, b, c, d\}$ can be four stacked twice with $i > 1$. Let e and f be the four stacks of Y_i. Consider the anchor set Y_{i-1} and without loss of generality assume that $Y_{i-1} = \{a, b, c, x\}$ The last bag that belongs to the cluster C_{i-1} then necessarily contains the vertices $\{a, b, c, x, d\}$. The subgraph G' induced by $\{a, b, c, d, e, f, x\}$ has the following property by maximality. The vertices a, b, c and d all have degree six in G' (i.e., they are connected to all other vertices of G'), hence G' contains $K_7^{-\triangle}$ as a subgraph and cannot be 1-planar by Lemma 5. □

Let G be a 1-planar 4-path and let $P = X_1, \ldots, X_l$ be an alternating path decomposition of G. Let $C_1, \ldots, C_s$ and $Y_1, \ldots, Y_s$ be the corresponding clusters and anchor sets, respectively, in the order given by P. By Properties 1 and 2 we have

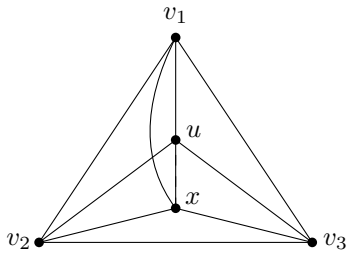


FIGURE 4.13: According to our rules, x is placed inside the triangle v_2v_3u such that xv_1 crosses v_2u

$|C_1| \in \{5,6\}$ and $|C_i| = 5$ for every $1 < i \leq l$. Let $v_1^1, v_2^1, v_3^1, v_4^1$ be the first, second, third and fourth vertex in Y_1, respectively, that is forgotten in this particular order. This implies in particular that $v_2^1, v_3^1, v_4^1 \in Y_2$. For every $1 < i < l$ let v_1^i, v_2^i, v_3^i be the vertices in $Y_i \cap Y_{i-1}$ that are forgotten first, second and third. Hence, $v_1^2 = v_2^1, v_2^2 = v_3^1$ and $v_3^2 = v_4^1$. Finally, we denote by x^i the vertex of $Y_{i+1} \setminus Y_i$ for every $1 \leq i < l$. In the following, we will provide an iterative construction of the final drawing Γ that is 1-planar - if we fail to construct Γ we claim that G is in fact not 1-planar, which will be shown in the subsequent lemma. The construction will maintain the following invariant.

Invariant 1. *Before inserting vertex x_i, the triangular region defined by $v_2^i v_3^i x^{i-1}$ is empty for any $1 \leq i \leq l$.*

Algorithm For the construction, we start off with a planar drawing of a K_4 induced by vertices $v_1^1 v_2^1 v_3^1$ and $v_4^1 = x^0$ such that v_4^1 is contained in the triangular region bounded by $v_1^1 v_2^1 v_3^1$. If $|C_1| = 6$, i.e., Y_1 is 4-stacked twice (which is the maximum amount by Property 1), let z be the remaining vertex in $C_1 \setminus (Y_1 \cup Y_2)$. We place z in the triangular region bounded by $v_1^1 v_2^1 v_4^1$ and route the edge zv_3^1 on the "outside", i.e., it intersects $v_1^1 v_2^1$ and stays in the outer-face of the initial K_4 until it reaches v_3^1. Afterwards, we place x^1 inside the triangle $v_2^1 v_3^1 x^0$ (recall that $x^0 = v_4^1$) such that the edge $x^1 v_1^1$ intersects the edge $v_3^1 v_4^1$. Otherwise, $|C_1| = 5$ and we consider the degree of x^1. If x^1 only has degree 5 in G (i.e., $x^1 \notin Y_3$) then we place it inside $v_1^1 v_2^1 x^0$ and route the edge $x^1 v_3^1$ such that it intersects $v_1^1 v_4^1$ which is fine since $v_1^1 \notin Y_2$ by construction. Finally, if the degree of x^1 is larger than five, we place it inside $v_2^1 v_3^1 x^0$ and route the edge $x^1 v_1^1$ such that it crosses $v_2^1 v_4^1$. Clearly, for every case of the initial step, Invariant 1 is maintained. Now for every $1 < i \leq l$ we place x^i inside the triangle $v_2^i v_3^i x^{i-1}$ such that $x^i v_1^i$ crosses $x^{i-1} v_2^i$ if possible and $x^{i-1} v_3^i$ otherwise; see Figure 4.13. By routing $x^i v_1^i$ according to our scheme, we can again guarantee that Invariant 1 is maintained as the region defined by $v_2^{i+1} = v_3^i$, $v_3^{i+1} = v_4^i$ and x^i remains empty.

Theorem 2. *Let G be a 4-path. Then, we can test in linear time whether G is 1-planar. In the affirmative case, the algorithm yields a 1-planar drawing Γ of G.*

Proof. Denote by Γ the drawing that we obtain by our construction. Clearly, if Γ is 1-planar, then it is a certificate for G. Hence it remains to show that if G is 1-planar, then the constructed drawing Γ according to our rules is 1-planar. For the sake of contradiction assume that the drawing Γ is not 1-planar. By Invariant 1 we have that the triangle $v_2^i v_3^i x^{i-1}$ is empty when drawing x^i for any $1 \leq i < l$. Hence only adding the edge $x^i v_1^i$ can violate 1-planarity. By our drawing rules this is the case if the triangles $\Delta_1^i = v_1^i v_2^i x^{i-1}$ and $\Delta_2^i = v_1^i v_3^i x^{i-1}$ are both non-empty.

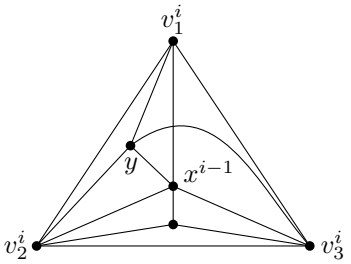


FIGURE 4.14: Vertex y is contained inside the triangle $v_1^i v_2^i x^{i-1}$

Let i be the smallest index such that Δ_1^i and Δ_2^i are both non-empty when we draw x^i. We distinguish the following two cases. Either there is a vertex inside Δ_1^i or Δ_2^i or they only contain edges. First assume there is a vertex y in $v_1^i v_2^i v_3^i$ outside of $v_2^i v_3^i x^{i-1}$. Without loss of generality assume that y lies inside Δ_1^i; see Figure 4.14. Since x^i is inserted directly after x^{i-1} by Property 2 the vertex y was necessarily inserted before x^{i-1}. Assume first that $i-1=1$. Then we have that $x^{i-1}=x^1$ and since y is in the interior of $v_1^i v_2^i v_3^i$, it follows that $y=x^0=v_4^1$. However, this is impossible as y is forgotten before $v_1^i=v_1^1$. Hence, $(i-1)>1$ and there exists a vertex u that is adjacent to v_1^i, v_2^i, v_3^i and y. Assume now that $(i-1)=2$ and thus $Y_1=\{v_1^i, v_2^i, v_3^i, u\}$. Since u is forgotten first, we have that $u=v_1^1$. Recall that y is contained inside $v_1^i v_2^i v_3^i$. By our construction rule and Property 1, this is only possible if either $|C_1|=5$ and the degree of y is larger than five or $|C_i|=6$. For the former case, we have that y needs to be connected to at least one more vertex besides $v_1^i v_2^i v_3^i, u$ and x^{i-1}. By Property 2 this means y would need to be part of the anchor set of Y_i, a contradiction. For the latter case, we have by construction that $y=x^1$ and there is a vertex z connected to v_1^i, v_2^i, v_3^i and u. We keep this fact in mind and take a step back to first consider the remaining case. If $(i-1)>2$, then we have $\{v_1^i, v_2^i, v_3^i, u\}=Y_j$ for some $1<j<i$. Then, there is a vertex $z\in Y_{j-1}$ adjacent to v_1^i, v_2^i, v_3^i, u. Thus, in both cases we observe the following configuration. Consider the subgraph G' induced by the vertices $\{z, u, y, v_1^i, v_2^i, v_3^i, x^{i-1}, x^i\}$. Vertices v_1^i, v_2^i and v_3^i have degree seven in G', vertices u,y and x^{i-1} have degree five in G' and vertices x^i and z have degree four in G', hence G' coincides with H, which is not 1-planar by Lemma 6.

It remains to consider the case where there is no vertex in $v_1^i v_2^i v_3^i$ outside of $v_2^i v_3^i x^{i-1}$. Then there are at least two edges that have an endpoint in $\{v_1^i, v_2^i, x^{i-1}\}$ and $\{v_1^i, v_3^i, x^{i-1}\}$, respectively and go through the corresponding triangle. But by construction, any such edge incident to v_1^i, v_2^i or v_3^i that passes through Δ_1^i or Δ_2^i necessarily requires an endpoint in either Δ_1^i or Δ_2^i and we are in the previous case. Since x^{i-1} contains at most one neighbor outside of the triangle $\{v_1^i, v_2^i, v_3^i\}$ by Property 2 and by construction, it follows that at most one of Δ_1^i or Δ_2^i does not contain a vertex but is still non-empty.

Clearly, the drawing can be constructed in linear time which concludes the proof. □

4.5 Pathwidth-3

In this section, we will provide approximation algorithms for testing whether a pathwidth-3 graph is k-planar. Let G be a pathwidth-3 graph and let $P=X_1, X_2, \ldots X_l$ of G be a path decomposition of G with clusters $C_1, \ldots, C_s$ and corresponding anchor

triplets $T_1, \ldots, T_s$. Our algorithm will construct the final drawing Γ of G iteratively, such that every intermediate drawing $\Gamma[: i]$ satisfies the following invariant.

Invariant 2. *Let C_j be the current cluster with anchor triplet $T_j = \{a, b, c\}$ and let $d \in X_i$ be a proper 2-stack connected to a and b. The region delimited by a, b and d in $\Gamma[: i]$ is either empty or it contains only other proper 2-stacks of a and b.*

Algorithm Denote by S the set of vertices with degree zero or one in G. We remove all vertices of S, compute a drawing Γ' for $G \setminus S$ which can then be completed to a drawing Γ of G without introducing additional crossings[1]. In particular, this implies that any vertex of a cluster which does not belong to an anchor triplet is a k-stack with $k \geq 2$. Consider the first bag $X_1 = \{a, b, c, d\}$ and assume that $d \notin T_1$. Then, we draw the vertices of X_1 such that a,b and c lie on the outer face, while d is contained inside this triangular region. In particular, assume first that d is a 2-stack. W.l.o.g. assume that d stacks a and b. Then, we place d ϵ-close to the edge ab, but still inside the triangular region defined by abc - clearly Invariant 2 is maintained as d is the first vertex which stacks a and b. If d is a 3-stack, we simply place it inside abc. Now that we have described the initial configuration, let us consider a vertex v_i that is introduced in bag X_i with $|X_i| = 4$. Assume that C_j is the current cluster with anchor triplet $T_j = \{a, b, c\}$. We distinguish between the following two cases.

1. v_i is a proper 2-stack of T_j

   W.l.o.g. assume that v_i is a 2-stack of $\{a, b\}$. Since Invariant 2 is maintained in $\Gamma[: i-1]$, there exists a vertex v' such that the region abv' is either empty or it only contains other 2-stacks of a and b. In particular, v' is also a 2-stack of a and b. We place v_i ϵ-close to v' such that abv_i completely contains the region abv' - hence $\Gamma[: i]$ maintains Invariant 2.

2. v_i is a proper 3-stack of T_j or $v_i \in T_{j+1}$.

   W.l.o.g. assume that $c \notin T_{j+1}$. Also, denote by v' the previous (proper) 3-stack of Y_j (if it exists). We place v_i inside abc such that it is completely contained inside the region delimited by abv', see Fig. 4.15. If v' does not exist, we simply place v_i inside abc. In particular, we observe that the edges ab_i and bv_i are crossing free in $\Gamma[: i]$ by construction.

Theorem 3. *Given a graph G of pathwidth 3, we can compute a 6-approximation to 1-planarity in linear time.*

Proof. Since G has pathwidth 3, we can compute in linear time [40] a path decomposition of width 3 and hence also an alternating path decomposition $P = X_1, X_2, \ldots X_l$ together with clusters $C_1, \ldots, C_s$ and anchor triplets $T_1, \ldots, T_r$.

We first observe that edges incident to a proper 2-stack are crossing free in Γ. Suppose the contrary and let d be a proper 2-stack of vertices a and b such that one of ad or ab is crossed by an edge e. Then, by Invariant 2, it follows that also ab, which belongs to the induced K_3 of an anchor triplet, is crossed by e. However, as the initial drawing is crossing free and, as shown in Case 2, the edges that belong to an anchor triplet are crossing free when introduced and remain crossing free when inserting 2- or 3-stacks by construction, we obtain a contradiction. Thus, every crossing we introduce occurs in Case 2. As $K_{3,7}$ is not 1-planar by Lemma 3, the latter case occurs at most 7 times, i.e., we have at most six proper 3-stacks and then have to change the anchor triplet. We can arrange them such that every edge is crossed at most six times, see Fig. 4.15 and hence we obtain the desired result. □

[1] degree one vertices will be placed epsilon close to their neighbors, while singletons are trivial

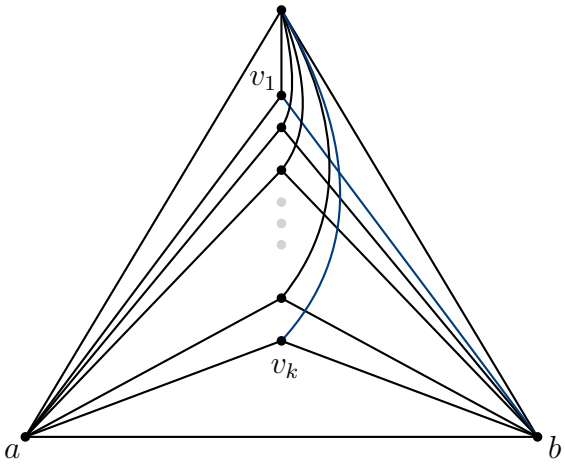


FIGURE 4.15: Illustration of k many 3-stacks of the anchor triplet such that every edge is crossed at most $k-1$ times. In particular, the dark blue edges are both crossed exactly $k-1$ times.

Theorem 4. *Given a graph G of pathwidth 3, the algorithm computes a $7k$-approximation to k-planarity in linear time.*

Proof. Assume that k is fixed. We construct our drawing exactly as in the proof of Theorem 3, hence the only interesting quantity is the number of 3-stacks for a fixed anchor triplet. In particular, if an anchor triplet is 3-stacked more than $7k$ many times, it follows by Lemma 2 that the given graph is not k-planar. Otherwise, we can construct a subdrawing of this cluster which is $7k$-planar as shown in Fig. 4.15 which concludes the proof. □

By using Lemma 3, we can obtain better constants for small k.

Corollary 1. *Given an input graph G of pathwidth 3, the algorithm computes a $6k$-approximation to k-planarity for $k \leq 4$ in linear time.*

Note that this approach also naturally extends to k-gap planarity.

Lemma 13. *$K_{3,13k+1}$ is not k-gap-planar.*

Proof. A k-gap planar graph with m edges can contain at most km many crossings. For $K_{3,a}$ we have $m = 3a$ and thus require

$$k3a \geq \cdot \lfloor \frac{a}{2} \rfloor \lfloor \frac{a-1}{2} \rfloor$$

which implies that $k \geq \lfloor \frac{a-1}{12} \rfloor$ which does not holds for $a \geq 13k+1$. □

Corollary 2. *Given an input graph G of pathwidth 3, the algorithm computes a $13k$-approximation to k-gap-planarity in linear time.*

4.6 w-Paths

For the remainder of this section let G be a w-path on n vertices – without loss of generality assume that $n \geq w+1$ and let $P = X_1, \ldots, X_l$ be a path decomposition of G. We remark here that for the case of 1-planar graphs, the 3- and 4-paths are

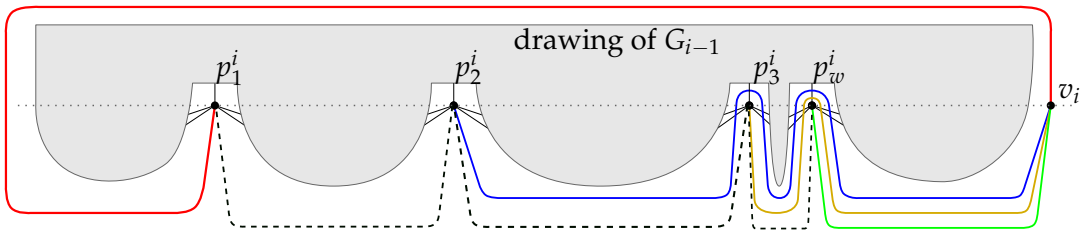


FIGURE 4.16: The construction of Biedl et. al. [170] to approximate the crossing number.

in a sense the only interesting cases of w-paths. In fact, since K_7 is not 1-planar[2], there is no w-path with $w \geq 6$ that is 1-planar. Regarding 5-paths, observe that a 5-path on 8 vertices has $25 > 4 \cdot 8 - 8$ edges, thus any 5-path that could be 1-planar can have at most 7 vertices. Since K_7 minus one edge is also not 1-planar [39], it follows that there exists exactly one 5-path that is 1-planar, namely the complete graph K_6. The following theorem will show that the construction of [170], which was used to obtain an approximation to the crossing number of w-paths (which is quadratic in the pathwidth), in fact yields a linear approximation (in the pathwidth) for k-planarity.

Before we state the theorem, we provide a brief description of the construction, refer to [170] for a more detailed discussion. Fig. 4.16 illustrates the construction. We arrange the vertices in an order $\{v_1, v_2, \dots v_n\}$ that is determined by their first appearance in P - while this order is well defined for the vertices which are not part of X_1, i.e., the vertices $v_{w+2} \dots, v_n$ we will order the vertices of X_1 such that v_1 is the first vertex of X_1 that is forgotten in P, v_2 is the second vertex of X_1 that is forgotten in P and so on.

Vertex v_j will be placed on coordinate $(j, 0)$. We will draw $G[X_1]$, which is a complete graph K_{w+1}, such that every edge is drawn as a half-circle above the x-axis. Assume that bag X_j introduces vertex v_i with $i \geq w + 2$. By construction, v_i is placed to the right of all other vertices that are placed so far. Denote by $p_1^i, \dots, p_w^i$ the neighbors of v_i in $G[: j]$ such that p_q^i is to the left of p_r^i if $q < r$. We can then observe two important properties.

One the one hand, we have that $\deg(p_1^i) \geq \deg(p_2^i) \geq \cdots \geq \deg(p_w^i)$. On the other hand, every neighbor of p_r^i in $G[: j]$ is also necessarily incident to p_q^i for $q < r$.

The edge $v_i p_1^i$ is routed counterclockwise around the drawing $\Gamma[: j]$, refer to the red edge in Fig. 4.16. The other edges are routed clockwise as a bundle, refer to the blue,gold and green edge in Fig. 4.16, each wrapping around the other vertices of X_j so that they remain visible from the bottom.

We are now ready to state the theorem.

Theorem 5. *Let G be a w-path on n vertices. We can compute in $\mathcal{O}(n)$ time a drawing Γ that is a $\mathcal{O}(w)$-approximation to k-planarity.*

Proof. Let Γ be the drawing of G obtained by this construction. Clearly, the drawing can be constructed in linear time. Let e be the edge of Γ that is involved in the most crossings. Assume that e is incident to vertex v_i and let X_j be the bag that introduces v_i. We will count the crossings of e in $\Gamma[: j]$ and in $\Gamma[j :]$ separately. Consider first the crossings of e in $\Gamma[: j]$. If $j = 1$, i.e., v_i belongs to X_1, then e is part of the initial K_{w+1}. By construction, e is crossed at most $\binom{w}{2}$ many times. Otherwise, if $j > i$, then edge e crosses the incident edges of at most $w - 1$ many vertices by

[2] which follows from a simple edge-density argument

construction. If the maximum number of such edges incident to one vertex u is x, then the number of crossings of e in $\Gamma[:j]$ is at most $(w-1)\cdot x \leq wx$. Since $\deg(p_a^i) \geq \deg(p_b^i)$ for any $a < b$ holds and since no edge wraps around p_1^i or p_2^i by construction, it follows that $u = p_3^1$. Since G is a w-path, every neighbor of p_3^i is also a neighbor of p_2^i and p_1^i – hence, we have a $K_{3,x-2}$ in G.

Consider now the crossings of e in $\Gamma[j:]$. Edge e is only crossed as long as v_i is part of the anchor set of the current cluster. Assume that e is crossed y times in $\Gamma[j:]$. For every time that e is crossed, there exists a *witness vertex* v_r such that v_i is p_l^r with $r > i$ and $l \geq 3$. Since every witness vertex contributes at most w many crossings to e, there are at least $\frac{y}{w}$ many witness vertices. Consider the last witness vertex v_s of e. Since $v_i = p_l^s$ for $l \geq 3$, p_1^s and p_2^s as well as v_i are adjacent to all $\frac{y}{w}$ many witness vertices of e – hence they form a $K_{3,\frac{y}{w}}$.

Let us now consider the approximation ratio. If $j = 1$, then an edge e is crossed $\frac{w}{2} + y$ times. Since G contains a K_{w+1}, we have that G is not k-planar for $k < \frac{w^2}{7.62^2}$ by Lemma 4. On the other hand, G contains a $K_{3,\frac{y}{w}}$ and thus G is not k-planar for $k < 7\frac{y}{w} + 1$ by Lemma 2. Hence, G is not k-planar for $k < \max(\frac{w^2}{7.62^2}, 7\frac{y}{w} + 1)$ and thus our $(\frac{w}{2} + y)$-planar drawing is a $c \cdot w$-approximation for a constant c. If $j > 1$, then e has at most $wx + y$ crossings and we have a $K_{3,x-2}$ and $K_{3,\frac{y}{w}}$ in G, hence G is not k-planar for $k < \max(7(x-2)+1, 7\frac{y}{w} + 1)$ and thus our $(wx + y)$-planar drawing is a $c \cdot w$ approximation again. □

4.7 Open problems

We conclude this chapter with some open problems that arise from our work.

1. Can the result of Section 4.3 be extended to all pathwidth-3 graphs? To this end, lets us provide a possible way to tackle the problem. Regarding the proper 2-stacks, we conjecture that every 1-planar pathwidth-3 graph admits a drawing Γ that satisfies Invariant 2, i.e., the region between a proper 2-stack v of an edge ab contains at most other proper 2-stacks of ab. We remark here that this property will not hold for arbitrary large pathwidth-k graphs as witnessed by Fig. 4.17.

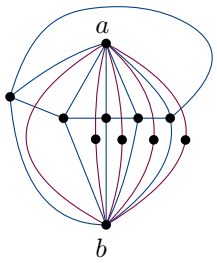


FIGURE 4.17: Edge ab and its proper 2-stacks are colored red. We cannot move the vertices outside of the region delimited by a, b and the right-most 2-stack of ab and still obtain a 1-planar drawing.

   If we assume that there always exists a drawing with this property, the only case which then differs from Section 4.3 is if a vertex v is a k-stack with $k \leq 2$, but not a proper k-stack. If we assume that bag X_i introduces v, then $G[X_i]$ is not a K_4, i.e., we can not apply Lemma 8 and hence there are more possibilities to 3-stack the anchor triplet which contains v.

2. Can we derive exact algorithms for w-paths analogous to Section 4.3 and Section 4.4 for small values of k-planarity, i.e., for $k = 2$? This would likely require additional results of small graphs which are not k-planar, which might be of independent interest themselves.

3. Can k-planarity be approximated by a constant $c \geq 2$?

Chapter 5

On Morphing 1-planar graphs

The notion of *topological equivalence* introduced in Chapter 2 gives rise to the following question: Assuming Γ_a and Γ_b are two topological equivalent drawings, i.e., they belong to the same combinatorial arrangement $\mathcal{A}$, is there a continuously changing family of drawings $\mathcal{F}_i$ such that $\mathcal{F}_0 = \Gamma_a, \mathcal{F}_1 = \Gamma_b$ and $\mathcal{F}_i \in \mathcal{A}$ for any $i \in [0,1]$? We refer to such a family as a *morph*. Remark that without additional restrictions on the (intermediate) drawings, the universal existence of such a morph can be shown as follows: Let $\mathcal{P}_a$ ($\mathcal{P}_b$) be the planarization of Γ_a and Γ_b, respectively. By definition, they have the same combinatorial embedding. The morph can then be constructed by applying Lemma 1.2 of [124] to every vertex. Hence, additional geometric restrictions are placed on the drawings which have to be maintained throughout the morph. The basic requirement is that the edges are drawn as straight-line segments throughout the morph, while additional geometric properties such as convexity or orthogonality are sometimes enforced.

Related work Seminal results date back to Cairns [51] who showed the existence of a morph between any two drawings of a triangulation, which was extended by Thomassen [171] to general planar graphs. Further results were concerned with the complexity of the morph [4, 52, 64] and applications in computer vision. However, most of the literature is restricted to morphing planar graphs - only a few results are known for graphs which admit embeddings on surfaces of higher (but bounded) genus such as the torus [52].

Our contribution We present the first morphing algorithm for graphs with non-constant genus. In particular, we show the existence of a morph between any pair of topologically equivalent graphs for a meaningful family of 1-plane geometric graphs by providing a constructive algorithm.
This chapter is based on joint work with Patrizio Angelini, Michael A. Bekos and Fabrizio Montecchiani and was published in the "Journal of Computational Geometry" [12].

5.1 Notation and Definitions

Before we state some specific notation that is used throughout the section, we start with a definition which is only used in this part of the thesis. The *BC-tree* $\mathcal{T}$ of a connected graph G represents the decomposition of G into its biconnected components, called *blocks*. Namely, $\mathcal{T}$ has a *B-node* for each block of G and a *C-node* for each cutvertex of G, such that each B-node is connected to the C-nodes that are part of its block.

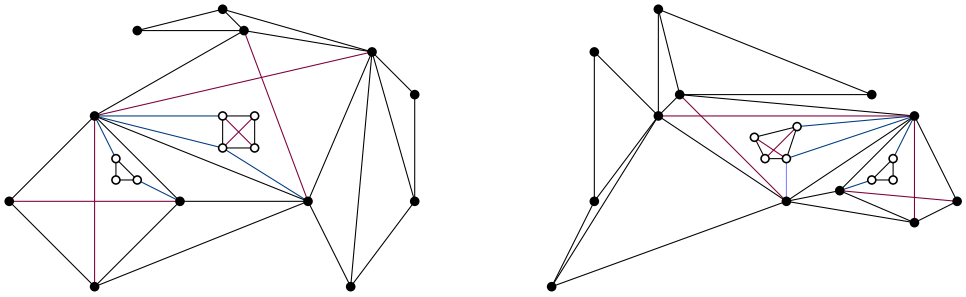


FIGURE 5.1: Two topologically equivalent kite-planar 1-planar drawings of the same graph.

We begin by describing a set of sufficient conditions under which any pair $(\Gamma_a(G), \Gamma_b(G))$ of topologically-equivalent drawings of a 1-plane graph G admits a morph using our algorithm. At a high level, we require that every edge is crossed at most once, any pair of crossing edges is enclosed in a 4-cycle whose edges are uncrossed and while such a 4-cycle can contain an arbitrary subgraph H in its interior, we require that all edges which connect the endpoints of the 4-cycle to H are uncrossed. To be more precise, consider a K_4. Up to a choice of the external cell, only two different 1-planar drawings of K_4 yield a different combinatorial arrangement, namely, a planar drawing whose external cell is a 3-cycle, and a 1-planar drawing such that the external cell is bounded by a 4-cycle. Let G be a 1-planar graph and let G' be an induced K_4 of G. If the external cell of $\Gamma(G')$ is bounded by a 4-cycle, then we call G' a *kite* in $\Gamma(G)$. The crossing-free edges which bound the external cell are called *kite-edges*, while the remaining two edges which cross in the interior are called *crossing edges*. The following relations between a kite K and a vertex u can occur:

1. u belongs to K, i.e., u is one of the four vertices that define K,
2. u is inside K, i.e., the vertex lies in the interior of the closed region defined by the 4-cycle bounding the external cell or
3. u is outside K.

A kite is *empty* if it encloses no vertex; otherwise, it is *non-empty*. An edge uv is a *binding edge* (blue in Figure 5.1) if u belongs to a non-empty kite K and v is inside K. We can now formally introduce *kite-planar* 1*-planar* drawings.

Definition 1. *A straight-line drawing is kite-planar* 1*-planar, or* 1*-kite-planar for short, if the following properties hold:*

(P.1) *every edge is crossed at most once,*

(P.2) *every pair of crossing edges induces a kite whose kite edges are uncrossed, and*

(P.3) *every binding edge is uncrossed.*

Moreover, these requirements allow us to partition the set of vertices into different levels. Let $\Gamma(G)$ be a 1-kite-planar drawing of G. A vertex v of G has *level* 0 if no kite encloses v. Otherwise, vertex v has level $i > 0$ if the maximum level of the vertices forming a kite that encloses v is $i - 1$. In Figure 5.1, the black (white, resp.) vertices are of level 0 (level 1, resp.).

The following property follows from **(P.3)** of Definition 1.

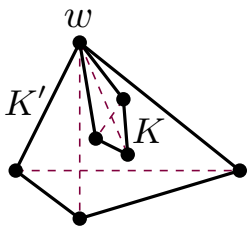


FIGURE 5.2: Illustration for the proof of Property 3: the case in which w belongs to K'.

Property 3. *If two vertices belong to the same kite of a* 1*-kite-planar drawing* $\Gamma(G)$*, then they are of the same level.*

Proof. Suppose for a contradiction that two vertices v and w which belong to a kite K are of different level, w.l.o.g. assume that the level of v is larger than the level of w. Then there exists a kite K' which encloses v but not w. Assume first that vw does not belong to K'. In this case, w is necessarily outside of K'. But then the edge vw that belongs to kite K is necessarily intersecting a kite edge of K' which violates **(P.2)** of Definition 1. Thus, w belongs to K', but then the crossed edge of K incident to w is a binding edge of K', a contradiction to **(P.3)** of Definition 1, see Fig. 5.2 for an illustration. It follows that v and w are of the same level which concludes the proof. □

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

Theorem 6. *There exists a morph between any pair of topologically-equivalent kite-planar* 1*-planar drawings.*

Theorem 27 implies that, for a fixed graph, the space of its topologically-equivalent kite-planar 1-planar drawings is connected. The proof is constructive, although the vertices may use trajectories of unbounded complexity.

Outline. Our proof will consist of a recursive construction based on the level ℓ of a vertex of G. The base case will construct a morph for the vertices of level $\ell = 0$, while the recursive case will describe the trajectory of the vertices of level $\ell > 0$ using the computed morph of lower level vertices. A schematized version of the algorithm can be found in Figure 5.4. To be more precise, we will compute a morph where the kite boundary of each kite will be drawn as a strictly-convex polygon, which will guarantee that, throughout the morph, the corresponding edges will stay inside the respective boundary. The main challenge is due to the fact that the interior of the kites is not empty - to overcome this issue, we will remove the interior of a kite (which are vertices of higher level), compute a morph which maintains the convexity, and afterwards suitably reinsert and morph the removed subdrawing.

Key ingredients. We will heavily use the following two ingredients for intermediate steps. The first result by Aronov et al. [18] establishes that one can *compatibly triangulate* two topologically-equivalent planar drawings of a planar graph, which will be used to guarantee that, after potentially removing the subgraphs inside the kites, the resulting regions are strictly convex.

Theorem 7 (Aronov et al. [18])**.** *Given two topologically-equivalent planar drawings* $\Gamma_a(P)$ *and* $\Gamma_b(P)$ *of the same n-vertex planar graph* P*, it is possible to augment* $\Gamma_a(P)$ *and* $\Gamma_b(P)$ *to two topologically-equivalent planar drawings* $\Gamma_a(P')$ *and* $\Gamma_b(P')$ *of the same maximal planar graph* P' *such that* $\Gamma_a(P) \subseteq \Gamma_a(P')$*,* $\Gamma_b(P) \subseteq \Gamma_b(P')$*, and the order of* $P' \setminus P$ *is* $O(n^2)$*.*

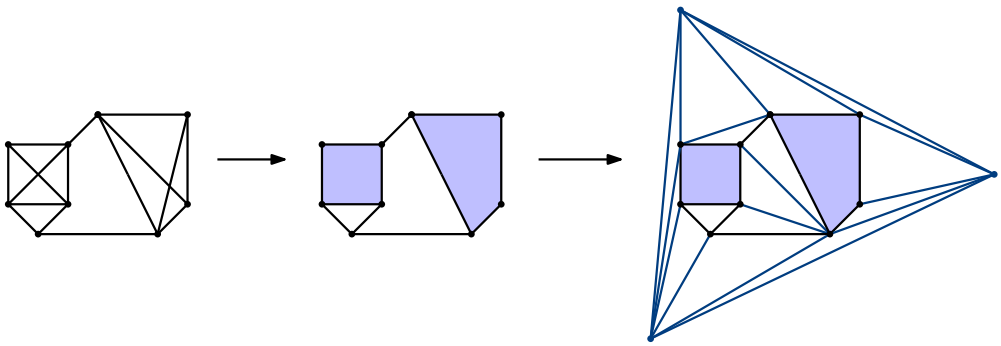


FIGURE 5.3: Illustration of the transformations $\Gamma_a(G) \rightarrow \Gamma_a(P) \rightarrow \Gamma_a(P')$; marked faces are light blue; edges of $P' \setminus P$ are dark blue.

The second ingredient is a result by Angelini et al. [14], which allows us to morph a pair of strictly-convex drawings by preserving the strict convexity of the faces. The main properties of this result are summarized in the next theorem.

Theorem 8 (Angelini et al. [14]). *Let $\langle \Gamma_a(P), \Gamma_b(P) \rangle$ be a pair of topologically-equivalent strictly-convex drawings of a planar graph P. There is a morph between $\Gamma_a(P)$ and $\Gamma_b(P)$ in which every intermediate drawing is strictly convex. In addition, if the outer face of G has only three vertices and each of them has the same position in $\Gamma_a(P)$ and $\Gamma_b(P)$, then these three vertices do not move during this morph.*

5.2 Base case

For the base case, we have that all vertices of G are of level $\ell = 0$, hence every kite of G, if any exist, are empty. Denote by P the graph obtained from G by removing both crossing edges for every kite. Let $\langle \Gamma_a(P), \Gamma_b(P) \rangle$ be the restrictions of $\langle \Gamma_a(G), \Gamma_b(G) \rangle$ to P, respectively. Clearly, $\langle \Gamma_a(G), \Gamma_b(G) \rangle$ is a pair of topologically-equivalent drawings of the plane subgraph P of G. Since the kite edges of each kite K of G are uncrossed by **(P.2)** of Definition 1, it follows that they bound a quadrangular face f_K in P, which will be called *marked*. We will now compatibly triangulate $\langle \Gamma_a(G), \Gamma_b(G) \rangle$ by applying Theorem 7 to $\langle \Gamma_a(P), \Gamma_b(P) \rangle$, except for the marked faces of P. Denote by P' the resulting graph and by $\langle \Gamma_a(P'), \Gamma_b(P') \rangle$ the resulting drawings. Clearly, we have that $\langle \Gamma_a(P'), \Gamma_b(P') \rangle$ are topologically equivalent. Moreover, every face in both drawings is either a quadrangle (if the corresponding face in P was marked) or a triangle.

We observe the following property regarding the convexity.

Observation 1. *The pair $\langle \Gamma_a(P'), \Gamma_b(P') \rangle$ consists of two strictly-convex drawings of P'. Moreover, P' is triconnected.*

Proof. The triangular faces of P' are trivially strictly convex. Thus let us consider the marked faces of P'. Every corresponding kite contains two crossing edges in G, thus the boundary is drawn strictly convex in both $\Gamma_a(G)$ and $\Gamma_b(G)$ and therefore also in both $\Gamma_a(P')$ and $\Gamma_b(P')$ as they remained untouched during the application of Theorem 7. It follows that every face of P' is strictly convex, which implies that P' is internally triconnected [55, Corollary 1]. Moreover, since the outer-face of P' is a 3-cycle, P' is in fact triconnected which concludes the proof. □

Since $\langle \Gamma_a(P'), \Gamma_b(P') \rangle$ is a pair of strictly-convex drawings of P' , we can apply Theorem 8 to compute a morph for them which maintains the strict convexity of

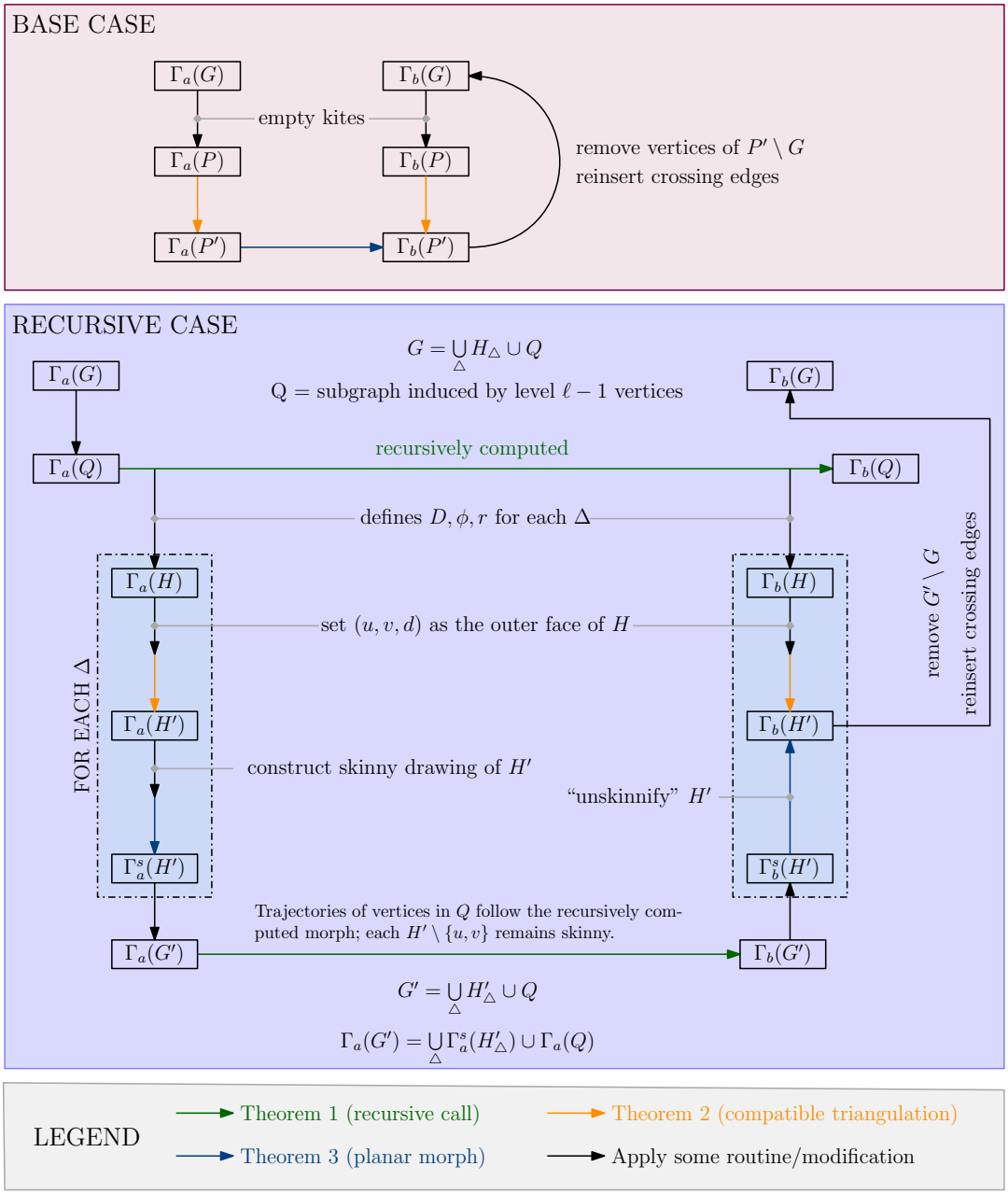


FIGURE 5.4: Schematic illustration for the proof of Theorem 6

the drawing at any time instant; in particular, this means that the marked faces stay strictly convex during the morph. Thus, we can reinsert the crossing edges of each kite and guarantee that they are fully contained inside the kite at any time instance. Augmenting P' with the crossed edges yields a supergraph of G, from which we can easily derive the morph of G by neglecting the vertices and edges which were introduced when applying Theorem 7. This concludes the base case of our morph, refer to the topmost (pink) box of Figure 5.4.

5.3 Recursive case

Here, we describe the recursive case of our construction where the maximum level ℓ of a vertex is strictly larger than zero. Refer to the middle (light blue) box of Figure 5.4 for an illustration of the main steps.

Let Q be the graph obtained by removing all the vertices of level ℓ from G, and let $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$ be the restriction of $\langle \Gamma_a(G), \Gamma_b(G) \rangle$ to Q. Clearly, the two subdrawings of Q are topologically equivalent; moreover, the maximum level of a vertex is $\ell - 1$. Hence, we can recursively compute a morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$ which prescribes the trajectories of the vertices of Q. Hence, in the remainder we have to find trajectories for the vertices of level ℓ which are compatible with the morph of Q. It is worth observing that the natural idea of a straightforward application of recursion to the subgraph contained in every triangular region defined by a kite that is empty in $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$ but not in $\langle \Gamma_a(G), \Gamma_b(G) \rangle$ does not work, since the algorithms in [14, 87] do not allow one to prescribe the trajectories of the vertices of the outer face, which are a subset of the vertices of Q, explicitly - and thus we can not guarantee that the morph of $\langle \Gamma_a(G), \Gamma_b(G) \rangle$ is compatible with the recursively computed morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$. Therefore, in what follows, we describe a more elaborated approach to overcome this issue. In particular, we describe how to incorporate the trajectories of the level-ℓ vertices into the morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$, so to obtain the desired morph of $\langle \Gamma_a(G), \Gamma_b(G) \rangle$.

5.3.1 Setting up the morph

Recall that by Property 3, a vertex of level ℓ lies inside a kite K whose vertices are all of level $\ell - 1$. Thus, K is empty in Q but not in G. The two crossing edges of G define four triangular regions (each bounded by two real vertices and the intersection point of the crossing edges). Since the morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$ guarantees that K remains strictly convex (the base case clearly has this property by Observation 1 and Theorem 8, while we will assure this in the following for the recursive case), it follows that the triangular regions remain non-degenerate throughout the morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$. We refer to each of these four triangular regions as a *piece of a kite*. Consider such a piece $\triangle$ of our kite K. The (unique) edge $uv \in G$ of $\triangle$ which belongs to the boundary of K is called the *base edge*. Since any piece of a kite remains non-degenerate, we have the following property.

Property 4. *Let $\triangle$ be a piece of a kite with base edge uv. There exists a half-disk D that, throughout the whole morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$, has the following properties (see also Figure 5.5 for an illustration):*

- *half-disk D lies in $\triangle$ and is centered at the midpoint w of uv, and*
- *the length of its radius is positive and it does not change.*

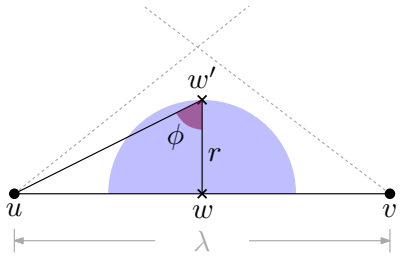


FIGURE 5.5: Illustration of the half-disk D of $\triangle$ and its geometric properties.

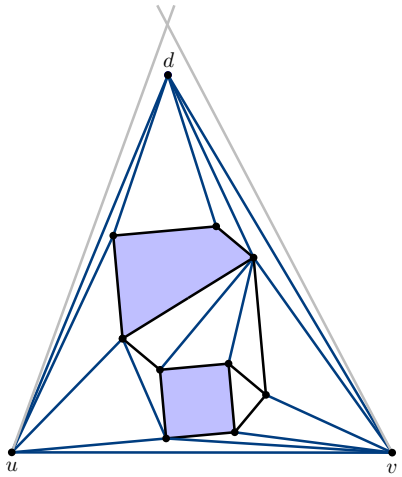


FIGURE 5.6: Illustration of the addition of dummy vertex d and the triangulation of H to obtain H'. Original edges are black.

We will now discuss how to define the length of the radius explicitly. Let λ be the smallest length of the base edge (u, v) during the morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$, let r be the radius of half-disk D perpendicular to (u, v), and let w' be the endpoint of r different from the midpoint w of (u, v). Also, denote by t^* any time instant of the morph when the length of (u, v) equals λ, and let ϕ be the internal angle at w' of the triangle formed by u, w and w' at time t^*. In particular, ϕ satisfies $\tan(\phi) = \frac{\lambda}{2} \cdot \frac{1}{|r|}$.

Consider now the subgraph $G \setminus Q$ which is induced by the level-ℓ vertices. In particular, let $H_\triangle$ be the subgraph of this graph that lies inside $\triangle$. The goal is to compute a drawing of $H_\triangle$ which is small enough to fit inside D but at the same time avoids the introduction of new crossings with the binding edges that connect u and v to $H_\triangle$. For ease of notation, from now on we will refer to $H_\triangle$ as H. In the first step, we have to augment H as well as its drawings in $\langle \Gamma_a(G), \Gamma_b(G) \rangle$ in order to leverage our tools. In particular, we add a dummy vertex d connected to u and v, which is placed sufficiently close to the crossing point of the two diagonal of K in both $\Gamma_a(G)$ and $\Gamma_b(G)$. This ensures that the triangular region formed by u,v and d properly contains H in both $\Gamma_a(G)$ and $\Gamma_b(G)$.

Next, in order to apply Theorem 8 at a later point, we want to create a strictly-convex drawing of (a supergraph of) H. Similar to the transformation from P to P' in Section 5.2, we remove the crossing edges of every kite of $H \cup \{u, v, d\}$, which are empty since any vertex contained inside such a kite would have level $\ell + 1$ and we mark the resulting quadrangular face. Then we apply Theorem 7 to the resulting planar subgraph of $H \cup \{u, v, d\}$ and to its drawings in $\langle \Gamma_a(G), \Gamma_b(G) \rangle$, except for its marked faces, see Fig. 5.6.

Let us denote the resulting graph as H'. By an analogous argument to Observation 1 we have that $\langle \Gamma_a(H'), \Gamma_b(H') \rangle$ is a pair of topologically-equivalent strictly convex drawings, and that H' is triconnected. We can further make the following observation.

Observation 2. *Every face incident to u or to v in H' is triangular.*

Proof. Any face in H' is either triangular, or quadrangular if marked and a marked face contains a pair of crossing edges in the original graph. Thus, if u or v would be incident to non-triangular faces, a binding edge of u or v would be crossed, a contradiction. □

We now consider the part of H' which will be redrawn (i.e., we will describe an appropriate morph) into D. Namely, consider the graph obtained from H' by removing u and v and denote by $\mathcal{C}$ the graph formed by the vertices and the edges of its outer face. The following lemma will establish some properties of this graph.

Lemma 14. *Graph $\mathcal{C}$ has the following properties:*

(i) *$\mathcal{C}$ is outerplane and connected.*

(ii) *Each block of $\mathcal{C}$ is a cycle, possibly degenerated to a single edge.*

(iii) *Every cutvertex of $\mathcal{C}$ is connected to both u and v in H'.*

(iv) *The BC-tree of $\mathcal{C}$ is a path.*

(v) *Every non-cutvertex of $\mathcal{C}$ is connected to exactly one of u and v in H', with the exception of exactly two vertices (one of them is d) which belong to the blocks of $\mathcal{C}$ corresponding to degree-1 B-nodes in the BC-tree of $\mathcal{C}$.*

Proof. By definition, $\mathcal{C}$ is formed by the vertices and edges of the outer face of $H' \setminus \{u, v\}$ and thus it is outerplane. Since H' is triconnected as established earlier, the removal of the two vertices u and v does not yield a disconnected graph, thus $\mathcal{C}$ is connected which establishes Property (i).

Property (ii) is then immediate, as $\mathcal{C}$ is a connected graph which corresponds to the outer face of $H' \setminus \{u, v\}$. In order to prove Property (iii), we will in fact show a slightly stronger statement. Namely, let c be a cutvertex of $\mathcal{C}$ and consider a walk δC along the boundary of the outer face of $\mathcal{C}$. We will show that any occurrence of c in δC implies the existence of one copy of uc or vc in H'. Assuming momentarily that this claim is true, since δC is a closed walk, we have that every cutvertex is visited at least twice and by simplicity of H' every cutvertex is visited exactly twice and the result then follows. In order to prove the claim, suppose a and b are two vertices immediately before and immediately after one occurrence of c in δC. Since c is a a cutvertex, a and b belong to different biconnected components of $\mathcal{C}$. We distinguish between the two following cases.

- There is a face f of H' that contains both a and b. Recall that by construction, every face of H' is either a triangle or a marked quadrangle. If f is a triangle, then f has exactly a,b and c on its boundary - but then the edge ab exists, which is a contradiction to c being a cutvertex. Otherwise, f is a marked quadrangle that contains a, b, c and some other vertex x on its boundary. By Observation 2, no marked quadrangle contains either u or v on its boundary, thus x belongs to $\mathcal{C}$, which is again a contradiction to the fact that c is a cutvertex.

- There is no face of H' that contains both a and b. Then, there exists an edge xc which belongs to H', such that xc does not belong to $H' \setminus \{u, v\}$ and hence not to $\mathcal{C}$. This means that x is either u or v as desired.

To show Property (iv) we will show that all $B-$ and $C-$nodes of the $BC-$tree have degree at most 2 which then implies that the BC-tree is in fact a path. To achieve this, we prove that each cutvertex of $\mathcal{C}$ belongs to exactly two blocks and each block contains at most two cutvertices. Recall that by Property (iii), every cutvertex occurs exactly twice in δC and thus belongs to exactly two blocks. For the other part, suppose for a contradiction that there exists a block that contains at least three cutvertices c_1,c_2 and c_3. The boundary of $\mathcal{C}$ then contains three vertex-disjoint paths that pairwise connect these cutvertices. But then since u and v are connected to all of c_1, c_2 and c_3 by Property (iii) and since uv also belongs to H', we have that H' contains a K_5-minor which contradicts the planarity of H'.

Finally, concerning Property (v), we first show the existence of two non-cutvertices of $\mathcal{C}$ which are connected to both u and v. By construction, vertex d is one of them. For the other, consider the internal face of H' which is incident to uv. Since no face incident to u or v is marked by Observation 2, it follows that the third vertex of this face, which we denote by d', is necessarily connected to both u and v. Hence, all cutvertices of $\mathcal{C}$ as well as d and d' are connected to both u and v. The remaining vertices of $\mathcal{C}$ are, by definition, on the outer face of $H' \setminus \{u, v\}$. Again, as every face incident to u or v is triangular by Observation 2, it follows that every such vertex is connected to exactly one of u and v. Lastly, if d or d' would belong to a degree-2 B-node of the BC-tree of C, then again the cutvertices of that B-node in addition to d or d' and u and v would form a K_5-minor analogous to the proof of Property (iv) which concludes the proof. □

Motivated by (ii) and (iv) of Lemma 14, we call $\mathcal{C}$ a *chain of cycles* from now on, where every block (even if degenerated to single edge) is a *cycle*. As specified in Property (v) of Lemma 14, we refer to the non-cutvertex of $\mathcal{C}$ which is adjacent to both u and v that is different from d as d'.

5.3.2 Maintaining visibility of $\mathcal{C}$ to u and v

Recall that the main challenge is to include the level-ℓ vertices into the recursively computed morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$. In this regard, we want to compute a preliminary morph of $\Gamma_a(H')$ to a strictly-convex drawing $\Gamma_a^s(H')$ that is *skinny*. On a high level, this drawing maintains the visibility's of the vertices of $\mathcal{C}$ to u and v which will be guaranteed by an appropriate choice of the angle ϕ with respect to the base edge uv, while it is small enough to fit into the disk D defined in Property 4. To be more precise, we state the following requirements.

R.1 Every cycle of $\mathcal{C}$ is drawn inside the disk D.

R.2 Every cycle of $\mathcal{C}$ is drawn strictly convex.

R.3 The cutvertices of $\mathcal{C}$, as well as d and d', lie on the radius r of D.

R.4 For every cycle of $\mathcal{C}$ and for every segment on its boundary, the smallest angle formed at the intersection of the line through r and the line through the segment is smaller than ϕ.

We will show the existence of such a drawing by an explicit construction which will leverage the properties of $\mathcal{C}$ described in Lemma 14.

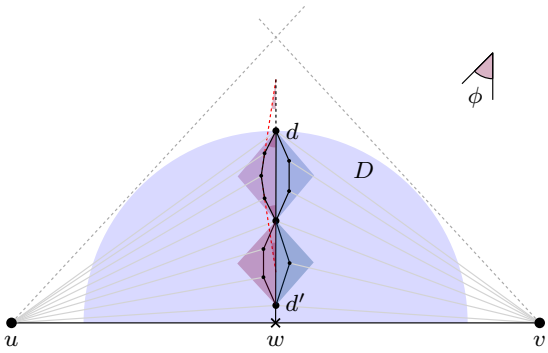


FIGURE 5.7: Illustration of the requirements R.R.1–R.R.4 of a skinny drawing.

Lemma 15. *There exists a drawing $\Gamma_a^s(H')$ that is strictly convex, skinny, and topologically equivalent to $\Gamma_a(H')$.*

Proof. We first give an description on how to compute the desired drawing $\Gamma_a^s(H')$. Throughout this description, we will show that $\Gamma_a^s(H')$ is skinny, i.e., requirements R.1–R.4 are met. In the end we will show that $\Gamma_a^s(H')$ is strictly convex and topologically equivalent to $\Gamma_a(H')$, which will then guarantee the existence of a morph between the two by Theorem 8. The construction is as follows.

- We place u and v in the same positions as they are in $\Gamma_a(H')$.
- We place all the vertices of $\mathcal{C}$ which are connected to both u and v, i.e., the cutvertices as well as d and d' on the radius r in the order they appear in δC such that d' is closer to the intersection point of r and uv than d. This obviously fulfills R.3.
- For a cycle c of $\mathcal{C}$, let x and y be the two vertices of c which are already placed on r. Define two isosceles triangles T_c^u and T_c^v sharing the same base $\overline{xy}$ such that the third vertex of each of them lies inside disk D on opposites sides of r and such that the internal angle at x and y is smaller than ϕ for each of them. For an illustration, refer to the colored triangles in Figure 5.7. In the following, we denote by H'_c the subgraph of H' induced by the vertices that lie on c or are contained inside c.
  - Recall that besides x and y, which are already placed on r, all remaining vertices of c are connected to exactly one of u or v. We construct a circular arc between x and y on each side of r which is completely contained inside T_c^u and T_c^v, respectively. The vertices of c that are connected only to u (only to v) are consecutive in δC and placed in this order equidistant on the circular arc that is completely contained inside T_c^u (T_c^v). This construction then immediately guarantees R.1, R.2, and R.4 for the drawing of c.
  - It remains to specify the position of the vertices of H'_c which are not part of $\mathcal{C}$. Here, we leverage the fact that since H'_c is drawn strictly convex in $\Gamma_a(H')$, it admits a strictly-convex drawing for any given strictly-convex drawing of its outer face [55]. Thus, we can apply the algorithm in [55] to construct a strictly-convex drawing of H'_c, whose outer face is exactly the drawing of c which we specified earlier.
- It remains to add the edges incident to u and v that are contained inside $\triangle$ to the resulting drawing. This completes the construction of $\Gamma_a^s(H')$.

We now prove that $\Gamma_a^s(H')$ is strictly convex and topologically equivalent to $\Gamma_a(H')$. By construction, every cycle c of $\mathcal{C}$ fulfills R.1–R.4, hence the drawing $\Gamma_a^s(H')$ is skinny. In particular, since R.3 is fulfilled, we have by Property 4 that no two edges incident to u or v intersect in $\Gamma_a^s(H')$. Since every subgraph H'_c is drawn strictly convex and since all faces incident to u and v in H' are triangular by Observation 2, it follows that $\Gamma_a^s(H')$ is strictly convex. To show topological equivalence, we remark that the algorithm in [55] as well as our construction maintain the cyclic order of edges around each vertex and, as previously observed, we do not introduce any crossing. Hence, $\Gamma_a^s(H')$ is topologically equivalent to $\Gamma_a(H')$ which concludes the proof. □

Now that we have established that $\Gamma_a^s(H')$ has all desired properties and that there exists a morph between $\Gamma_a(H')$ and $\Gamma_a^s(H')$, we need to specify this morph explicitly as we have some constraints that we can not ignore. Since both drawings are strictly convex and topologically equivalent, the preconditions of the first part of Theorem 8 are met. However, in order to perform this morph independently for each piece of a kite, we need to guarantee that the vertices u and v do not move and that all vertices of H' stay inside $\triangle$ throughout the whole morph. While u and v have the same position in $\Gamma_a(H')$ and $\Gamma_a^s(H')$ by construction, the position of d (which completes the triangular outer face) may differ. To overcome this issue, we introduce a new vertex d^* and place it at the same position in both $\Gamma_a(H')$ and $\Gamma_a^s(H')$. In particular, d^* is placed in the outer face of H' and it is connected to u,v and d. Clearly, the triangle defined by u, v and d^* contains all vertices of H' (as previously, the outer face was bounded by u,v and d) and the edges can be drawn crossing-free. After this augmentation, the requirements for the second part of Theorem 8 are fulfilled, which allows us to compute the desired morph of $\langle \Gamma_a(H'), \Gamma_a^s(H') \rangle$, after which we remove d^* from both drawings.

5.3.3 Performing the global morph

By applying the aforementioned procedure for each piece of a kite, we obtain a drawing $\Gamma_a(G')$ of the supergraph G' of G which is the union of Q and the augmented graphs $H'_\triangle$ that correspond to every piece of a kite $\triangle$. Remark that in order to construct such a drawing for each piece of a kite, we have to derive some parameters (such as the angle ϕ) from the shortest length of each base-edge throughout the morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$. In order to compute this length, we observe that the only operation which changes the edge-length of our drawing is the application of Theorem 8. In fact, the underlying algorithm is composed of a linear number of uni-directional morphs, where all the vertices move along parallel straight-line trajectories at each step. Thus it is straightforward to compute the shortest length of each base-edge. Hence, we can now describe the complete morph. Recall that $\Gamma_a(G')$ is composed of $\Gamma_a(Q)$ and the skinny drawing $\Gamma_a^s(H'_\triangle)$ of every graph $H'_\triangle$. By construction, the vertices of Q follow the same trajectory as in the morph between $\Gamma_a(Q)$ and $\Gamma_b(Q)$, which has been recursively computed. Hence it remains to describe the trajectory of the level-ℓ vertices. Since the level-ℓ vertices of each subgraph $H'_\triangle$ will remain inside $\triangle$ throughout the morph, we can consider them independently for each piece of a kite. Thus, we will describe the trajectories for one specific subgraph H' in the following. We will reuse the notation introduced in Sections 5.3.1 and 5.3.2.

The trajectories of u and v are specified by the morph of $\langle \Gamma_a(Q), \Gamma_b(Q) \rangle$, hence we only describe the trajectories of $H' \setminus \{u, v\}$ which are exactly the vertices of level-ℓ if H', see Fig. 5.8 for an example. The drawing of $H' \setminus \{u, v\}$ is a *rigid* copy of the

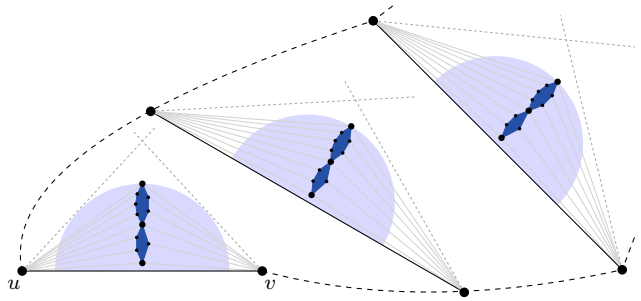


FIGURE 5.8: Computing the trajectories for the vertices of $H' \setminus \{u, v\}$ based on the, already computed, trajectories of u and v. The disk is highlighted in light blue, while the skinny drawing of $H' \setminus \{u, v\}$ is dark blue

skinny drawing $\Gamma_a^s(H' \setminus \{u, v\})$ rotated and translated such that the cutvertices of $\mathcal{C}$ as well as d and d' lie on the radius r of D perpendicular to the base edge uv and the distance between the intersection of r and uv and d' is the same as in $\Gamma_a^s(H')$. Clearly, the drawing of H' remains skinny, planar and strictly convex at every time instant by Section 5.3.2.

Let $\Gamma_b(G')$ be the drawing of G' obtained so far. It remains to undo some of the steps in order to obtain a drawing of the base graph. In particular, we need to transform, for each subgraph $H'_\triangle$, the current skinny drawing $\Gamma_b^s(H'_\triangle)$ in $\Gamma_b(G')$ to $\Gamma_b(H'_\triangle)$. By construction, $\Gamma_b^s(H'_\triangle)$ and $\Gamma_b(H'_\triangle)$ are topologically equivalent and strictly convex. Similarly as for $\Gamma_a(H'_\triangle)$, we ensure that the outer face of $H'_\triangle$ is drawn the same in both $\Gamma_b^s(H'_\triangle)$ and $\Gamma_b(H'_\triangle)$ by adding an extra vertex (see Section 5.3.2, where we introduced the auxiliary vertex d^* to achieve this property), which allows to apply Theorem 8 independently for each $H'_\triangle$. The target drawing $\Gamma_b(G)$ is then obtained by removing the vertices and edges in $G' \setminus G$ and by reinserting the crossed edges in the marked faces. This concludes the proof of Theorem 6.

5.4 Implications of Theorem 6

Here, we consider some known families of 1-planar graphs which admit 1-kite-planar drawings and therefore we can apply the aforementioned result to them.

By [38], we know that an n-vertex 1-planar graph can have at most $4n - 8$ edges. 1-planar graphs which achieve this bound are called *optimal*. Any 1-planar drawing of such a graph has the following two properties. The uncrossed edges induce a plane triconnected quadrangulation P and each pair of crossing edges of G is drawn inside a corresponding face of P, refer to [166]. Contrary to planar graphs, however, a corresponding result to [100] does not hold for 1-planar graphs, i.e., not every 1-planar graph can be drawn straight-line. In fact, the maximum number of edges is reduced to $4n - 9$ by this restriction [73]. An optimal 1-planar straight-line drawing is, similar to the general case, one in which the uncrossed edges induce a plane triconnected graph whose inner faces are all quadrangles, while the outer face is a triangle [73]. Clearly, this implies that each kite is empty which implies that the binding edges can not be crossed (since they do not exist) and every kite edge is present and uncrossed. Thus, any optimal 1-planar straight-line drawing is 1-kite-planar.

Another notable family of 1-planar graphs are *IC-planar* graphs [5, 46, 63, 143]. In an IC-planar drawing, no vertex can be incident to two crossed edges - conversely, the

crossed edges induce a matching. It follows that both the binding edges and the kite edges are necessarily uncrossed. Hence, if the drawing is in a sense maximal, i.e., all the kite edges are present, it is a 1-kite-planar drawing. The class of drawings where all kite edges are present is referred to *kite-augmented* [44] or *locally maximal* [90]. Hence, we obtain the following corollary to Theorem 27.

Corollary 3. *There exists a morph between any pair of topologically-equivalent straight-line drawings that are optimal* 1*-planar or kite-augmented IC-planar.*

5.5 Open problems

We conclude by stating some open problems.

1. Can our algorithm be extended to all 1-planar graphs? We remark here that by relaxing the requirement that the binding edges are uncrossed, we cannot easily morph the interior of the kites independently anymore, hence additional ideas will be required.

2. Can we bound the complexity, i.e., the number of linear steps which describe the vertex trajectories?

3. What about morphing other beyond-planar graph classes? One natural candidate would be the class of *h-framed drawings*. A drawing $\Gamma(G)$ is called h-framed, if after the removal of all crossing edges from $\Gamma(G)$, the remaining drawing $\Gamma(G')$ is a simple and biconnected planar graph that spans all vertices of G with faces of size at most h. Clearly, a 3-framed graph is a planar graph and every 4-framed graph is kite-planar (observe that binding edges are impossible in 4-framed graphs). It is an interesting open question whether our algorithm can be extended to the class of 5-framed graphs. We conjecture that this extension is at least possible for optimal 2-planar graphs, which form a subset of 5-framed graphs by [29].

We on purpose skipped over the elephant in the room so far.

4. Is it possible to morph all non-planar graphs?

To this end, let us consider the rather natural approach of finding a morph between two (topologically-equivalent) planarizations. The following short exhibition will show that the additional constraint that two opposite edges of an introduced dummy vertex have to maintain the same slope throughout the morph cannot be (easily) incorporated into the basic morphing paradigms introduced by Cairn [51] or Floater and Gotsman [91], respectively.

5.5.1 Algorithm of Cairn

Let G be a planar drawing and $\langle \Gamma_a(G), \Gamma_b(G) \rangle$ be two topologically-equivalent planar drawings of G. The following description of Cairn's algorithm is supposed to give a rough idea and omits several technical details to ease the presentation. First, we can observe that in any planar graph, hence in particular in G, we can identify a vertex u of maximum degree five, which we will contract to a neighbor v of u[1] such that v is still "visible" from its neighbors, i.e, the straight-line segment connecting

[1] in fact, if we place u directly on top of v this is not a proper morph, but here we just assume sufficiently close

the new position of v to its neighbor is not intersecting any other edge of the drawing. By induction, the resulting graph G' is planar and contains $(n-1)$ vertices and we use the induction hypothesis to derive that there exists a morph of G'. Hence, the morph of G can be described as follows.

1. Contract the edge uv (which can be done as a linear morph of only u) to obtain G'
2. Compute the morph from $\Gamma_a(G')$ to $\Gamma_b(G')$.
3. Compute a morph from the final drawing $\Gamma_b(G')$ to $\Gamma_b(G)$ in order to expand the edge uv again[2]

Let us now consider the application of the algorithm to a planarization $\mathcal{P}$. Since there are non-planar graphs with minimum degree larger than five, the only vertices of $\mathcal{P}$ of a suitable degree are in fact dummy vertices. Let us therefore consider the contraction of such a dummy vertex u with neighbors $v_1, \dots v_4$ such that v_1v_3 crosses v_2v_4 in the original drawing (of course it can be possible that one of $v_1, \dots v_4$ is a dummy vertex as well). W.l.o.g. we contract u to v_1. While one could potentially find a position sufficiently close to v_1 such that the slopes of (v_1, u) and (u, v_3) stay the same, the slopes of (v_2, u) and (v_4, u) will necessarily differ, thus making this rather straightforward approach infeasible.

5.5.2 Algorithm of Floater and Gotsman

The approach of Floater and Gotsman gives an explicit formula to compute the vertex coordinates at any (discrete) timestep t. To this end, let us make some additional assumptions, namely assume that the planarizations are 3-connected, that the boundary (i.e., the drawing of the outer face) of Γ'_a and Γ'_b is identical and that in Γ'_a and Γ'_b, any face is drawn convex and two faces share at most a common vertex or a common edge. We call such an embedding a *tiling*.

Under these assumptions, we can try to leverage the result of [91] who showed that there always exist a morph between Γ'_0 and Γ'_1 which guarantees that, at any time instance $t \in [0, 1]$, Γ'_t is a tiling and has the same combinatorial embedding as Γ'_0 and Γ'_1 which can be expressed by a convex combination of coefficients. Let us try to find a suitable choice of coefficients such that any two non-consecutive segments (w.r.t. the circular order) that are incident to a dummy vertex maintain the same slope throughout the morph. In [92], Floater extended the proof of correctness of Tutte [175] to arbitrary convex positions, i.e., for a vertex u with neighbors $(v_1, ..., v_k)$ we set

$$u = \sum_{j=1}^{k} \lambda_j v_j \tag{5.1}$$

such that

$$\sum_{j=1}^{k} \lambda_j = 1 \tag{5.2}$$

and

$$\lambda_j > 0 \tag{5.3}$$

[2] this last part can be tricky, as u contracted to v might not be visible from its other neighbors in Γ_1, hence v has to be chosen carefully and might require an initial morph to *convexify* the polygon which contains u in the inital drawing

Hence, if our choice of coefficients satisfies Eq. (5.1)-Eq. (5.3) we are guaranteed that we have a valid morph from Γ'_0 to Γ'_1. We will make use of the following lemma.

Lemma 16. *If point p lies on the intersection between the two segments p_1p_3 and p_2p_4, there exists $x, y \in (0,1)$ such that*

$$p = \frac{1}{2}(xp_1 + (1-x)p_3) + \frac{1}{2}(yp_2 + (1-y)p_4) \tag{5.4}$$

holds.

Proof. Since p lies on the intersection of the two segments, p lies on p_1p_3 and on p_2p_4. Hence, we can express p such that

$$p = xp_1 + (1-x)p_3$$
$$p = yp_2 + (1-y)p_4$$

holds. Summing both equations we get $2p = xp_1 + (1-x)p_3 + yp_2 + (1-y)p_4$ and therefore $p = \frac{1}{2}(xp_1 + (1-x)p_3) + \frac{1}{2}(yp_2 + (1-y)p_4)$ as desired. □

Let $t \in [0,1]$ be an arbitrary time instance. Let $u \in \mathcal{P}$ be a dummy vertex, i.e., u corresponds to a crossing in Γ_t. Clearly, $deg(u) = 4$. Let (u_1, u_2, u_3, u_4) be the neighbors of u. Our goal is to guarantee that segments u_1u and u_3u as well as u_2u and u_4u have the same slope in Γ'_t. Consider now Γ'_0. Note that the following construction can be seen as analogous to [91] by restricting the technique to *degenerate* triangles. The points (u_1, u_2, u_3, u_4) form a star-shaped polygon that contains u in its kernel. Let L_i be the line defined by u_i and u. Clearly, L_i also contains u_{i+2} (indices taken modulo 4). For any such line L_i (which can be interpreted as a degenerate triangle), we can express u as the convex combination of u_i and u_{i+2}. Namely,

$$u = \tau_1^0 u_1 + (1-\tau_1^0)u_3$$
$$u = \tau_2^0 u_2 + (1-\tau_2^0)u_4$$
$$u = (1-\tau_1^0)u_3 + \tau_1^0 u_1$$
$$u = (1-\tau_2^0)u_4 + \tau_2^0 u_2$$

Note that $\tau_1^0, \tau_2^0 \in (0,1)$. Now we define λ_j to be the average of how much u_j contributed to the four equalities and we get

$$\lambda_1 = \frac{1}{4}(\tau_1^0 + 0 + \tau_1^0 + 0) = \frac{\tau_1^0}{2}$$
$$\lambda_2 = \frac{\tau_2^0}{2}$$
$$\lambda_3 = \frac{(1-\tau_1^0)}{2}$$
$$\lambda_4 = \frac{(1-\tau_2^0)}{2}$$

Clearly, $\lambda_i > 0$ and $\sum_i \lambda_i = 1$ is satisfied.

Similarly, we can repeat the procedure for Γ'_1 and obtain $\lambda_1 = \frac{\tau_1^1}{2}$, $\lambda_2 = \frac{\tau_2^1}{2}$, $\lambda_3 = \frac{(1-\tau_1^1)}{2}$ and $\lambda_4 = \frac{(1-\tau_2^1)}{2}$.

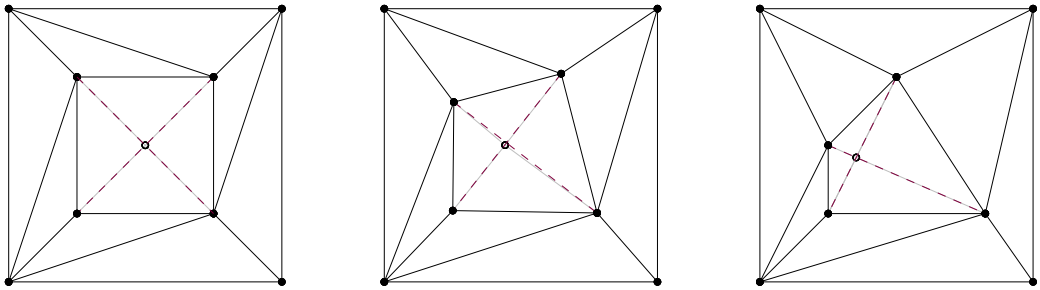


FIGURE 5.9: The left and the right drawing form a pair of topologically equivalent drawings. While the dummy vertex (circle) lies on the intersection of the lines for $\lambda = 0$ (left) and $\lambda = 1$ (right), this does not occur for $\lambda = 0.4$ (middle). Even though this difference is quite small, it is still sufficiently big so that its cause cannot be attributed to numerical errors.

Now, the lambda coefficients at time instance t satisfy $\lambda_i^t = (1-t)(\lambda_i^0) + t(\lambda_i^1)$ [91]. Hence, we express u^t as

$$u^t = \lambda_1^t u_1^t + \lambda_2^t u_2^t + \lambda_3^t u_3^t + \lambda_4^t u_4^t \tag{5.5}$$

which is equivalent to

$$u^t = \frac{1}{2}((1-t)\tau_1^0 + t\tau_1^1)u_1^t + \frac{1}{2}((1-t)\tau_2^0 + t\tau_2^1)u_2^t + \\ \frac{1}{2}((1-t)(1-\tau_1^0) + t(1-\tau_1^1))u_3^t + \frac{1}{2}((1-t)(1-\tau_2^0) + t(1-\tau_2^1))u_4^t \tag{5.6}$$

By multiplying the last equation with 2 and rearranging we obtain

$$2u^t = ((1-t)\tau_1^0 + t\tau_1^1)u_1^t + (1-t)(1-\tau_1^0) + t(1-\tau_1^1))u_3^t + \\ ((1-t)\tau_2^0 + t\tau_2^1)u_2^t + ((1-t)(1-\tau_2^0) + t(1-\tau_2^1))u_4^t \tag{5.7}$$

Set $x = (1-t)\tau_1^0 + t\tau_1^1$ and $y = (1-t)\tau_2^0 + t\tau_2^1$ and we obtain

$$2u^t = xu_1^t + (1-x)u_3^t + yu_2^t + (1-y)u_4^t \tag{5.8}$$

$xu_1^t + (1-x)u_3^t$ expresses a point P_1 which lies on the segment $u_1^t u_3^t$. Similarly, $yu_2^t + (1-y)u_4^t$ expresses a point P_2 which lies on the segment $u_2^t u_4^t$. Hence, $u = \frac{P_1+P_2}{2}$ is the midpoint of P_1 and P_2 - unfortunately, it does not necessarily hold that $P_1 = P_2$ holds as shown in Fig. 5.9, thus the approach is infeasible.

Chapter 6

Low degree RAC graphs with few bends

Cognitive experiments show that planar drawings, in particular straight-line drawings of graphs, are desirable since they do not introduce crossings nor curves of high-complexity. If no such drawing is possible, i.e., crossings cannot be avoided, the user study of [120] establishes that large crossing-angles are preferred. This motivates the class of k-bend RAC drawings, where every edge is a poly-line with at most k bends and the (smallest) angle of any two intersecting segments is right, i.e., the two crossing segments are perpendicular to each other. Since not all graphs admit k-bend RAC drawings for $k \in \{0, 1, 2\}$, it is natural to consider the containment of bounded degree graphs for these values of k.

Related work In [15], the authors showed that *Hamiltonian* degree-3 graphs are RAC. This result can be used, for example, to show that there exists RAC graphs which are not k-planar for any fixed k. Namely, n-vertex cube-connected cycles graphs are cubic and Hamiltonian[105] and therefore RAC, but they have crossing number $\Omega(n^2)$ [167] - hence in any drawing, there exists an edge with $\Omega(n)$ many crossings. The work of [75] showed that the complete bipartite graph $K_{4,4}$ does not admit a RAC drawing, hence there exists degree-4 graphs that are not RAC. Further, if one allows the presence of bends, it can be shown that every degree-3 graph is 1-bend RAC [13] and every degree-6 graph is 2-bend RAC [13]. If we allow even more bends per edge, the introductory paper of the class of k-bend RAC graphs [72] already established that every graph admits a 3-bend RAC drawing. Thus, subsequent research focused on the required drawing area, which was further motivated by the fact that (0-bend) RAC drawings might require exponential area since their recognition is hard in the existential theory of the reals [161]. The best result for the area of 3-bend RAC drawings is cubic (in the number of vertices) due to [95]. Subcubic area can be obtained if we allow up to six bends per edge [155].

Our contribution In Section 6.2, we will show that all 3-edge colorable degree-3 graphs admit a RAC drawing in quadratic area which can be computed in linear time. In Section 6.3, we provide two conceptually different algorithms that, for every degree-4 graph, compute a 1-bend RAC drawing in linear time that uses quadratic area. Finally, in Section 6.4, we provide a linear time algorithm which computes a 2-bend RAC drawing of quadratic size for any degree-8 graph.
Section 6.2 and Section 6.3 are based on joint work with Patrizio Angelini, Michael A. Bekos, Julia Katheder and Michael Kaufmann and was published at the "47th International Symposium on Mathematical Foundations of Computer Science (MFCS)" [8]. Section 6.4 is based on joint work with the aforementioned authors in addition

to Torsten Ueckerdt which is accepted at the "European Symposium on Algorithms (ESA) 2023".

6.1 Preliminaries

We assume in the following that G is connected, as otherwise we will apply the following algorithms to every component of G independently which will be easily merged together (i.e., drawn side by side) in the resulting drawing. The algorithms use a decomposition of the edge-set of G for their constructions. In particular, a 2-*factor* of an undirected graph G is a spanning subgraph of G which consists of vertex disjoint cycles. For a bounded degree graph, we have the following result available.

Theorem 9 (Eades, Symvonis, Whitesides [83])**.** *Let G be an undirected graph with maximum degree Δ and let $d = \lceil \Delta/2 \rceil$. Then, there exists a directed multigraph G' with the following properties:*

1. *each vertex of G' has indegree d and outdegree d;*
2. *G is a subgraph of the underlying undirected graph of G'; and*
3. *the edges of G' can be partitioned into d edge-disjoint directed 2-factors.*

The directed graph G' and the d 2-factors can be computed in $\mathcal{O}(\Delta^2|V[G]|)$ time.

Let F be a 2-factor of G computed by Theorem. 9. We will define a total order $\prec_F$ of the vertices of G based on F such that, for each cycle $C \in F$, the vertices appear consecutive in $\prec_F$ according to some traversal of F. This implies that any two vertices, which are adjacent in C, are consecutive in $\prec_F$ besides the first and the last vertex of the traversal of C. We will call the edge between these two vertices the *closing edge* of C. By construction, $\prec_F$ also implies a total order of the cycles of F. Consider an edge uv in $E \setminus F$ and let C and C' be the two cycles of F which contain u and v, respectively. If $C = C'$, i.e., u and v both belong to the same cycle $C \in F$, then uv is called a *chord* of C. Otherwise, if $C \neq C'$ and $C \prec_F C'$ holds, then uv is called a *forward edge* for u and a *backward edge* for v.

In order to control the crossing angles in the resulting drawing Γ, all crossings that occur in our constructions will consist of pairs of edge-segments (s_1, s_2) such that s_1 is parallel to the x-axis, while s_2 is perpendicular to the x-axis. This restricted class of RAC drawings is formally introduced in Chapter 7, where natural questions such as the containment relationship as well as edge-density bounds are considered. Motivated by this restriction, we can distinguish between the following type of incidence-relations between a vertex u and an edge uv. The edge uv uses an *orthogonal port* at u if the segment of uv incident to u is either horizontal or vertical (i.e., it could be involved in a crossing in our restricted setting), otherwise uv uses an *oblique port*. We will use a natural naming scheme for the orthogonal ports at u, i.e., we say that the segment of uv uses the N, E, S or W port at u if this segments is above, to the right, below or to the left of u.

6.2 RAC drawings of 3-edge-colorable degree-3 graphs

In this section, we will show that 3-edge-colorable degree-3 admit a RAC drawing of quadratic area that can be computed in linear time under the assumption that the 3-coloring of the edges is presented as part of the input, as it is NP-complete to test if even a 3-regular graph admits such a coloring by [114].

Theorem 10. *Given a* 3*-edge-colorable degree-*3 *graph G with n vertices and a* 3*-edge-coloring of G, it is possible to compute in $O(n)$ time a RAC drawing of G with $O(n^2)$ area.*

We can assume w.l.o.g. that G does not contain degree-1 vertices. Namely, assume that v is a degree-1 vertex in G connected to vertex u such that uv is assigned color one in the given 3-edge coloring. Then, we can substitute v with a 3-cycle (v_1, v_2, v_3) such that v_1 is connected to u, the color of v_1v_2 is two, the color of v_2v_3 is one and the color of v_3v_1 is three. Clearly, by applying this to every degree-1 vertex, we obtain a 3-edge colorable graph with asymptotically the same number of vertices (in particular, we introduce two new vertices for every degree-1 vertex in G).

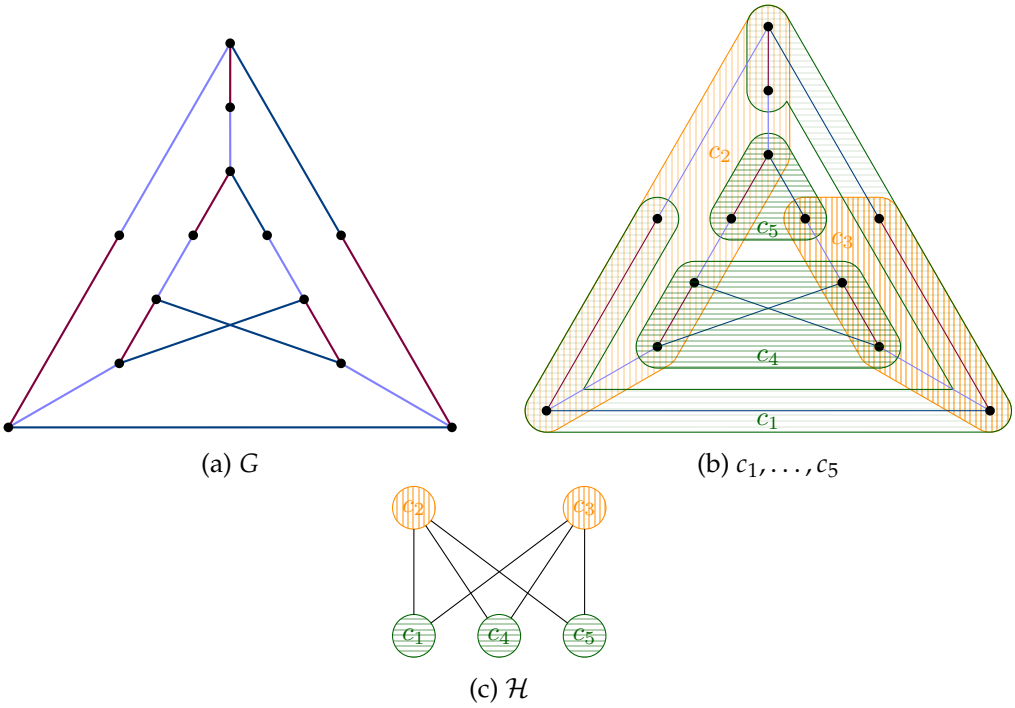
Since G is 3-edge-colorable, we can decompose the edge set of G into three matchings M_1, M_2 and M_3. In the final drawing Γ, the edges of M_1 will be drawn vertical, the edges of M_3 horizontal and the edges of M_2 will be drawn crossing free thus they do not require a particular slope. Denote by H_x and H_y the subgraphs of G induced by $M_1 \cup M_2$ and $M_2 \cup M_3$, respectively. By the initial observation we have that every vertex has degree at least two, which by definition belong to two different matchings, thus H_x and H_y are both spanning subgraphs of G. Moreover, as H_x and H_y are degree-two subgraphs of G, every component of H_x (H_y) is either a path or an even-length cycle (as an odd-length cycle would imply that two consecutive edges in a traversal of the cycle would belong to the same matching which clearly violates the matching property). In a sense, each of H_x and H_y is a subgraph of a 2-factor of G, however, the crucial difference is that they are not edge-disjoint as they share the edges of M_2.

The main idea of our algorithm is as follows: We will work through the components of H_x and H_y and place them consecutively in x- or y-direction. The main difficulty here is how to handle the closing edges for a cyclic component, as this edge spans several coordinates. In order to guarantee that this edge is crossing free, we have to consider a more elaborate strategy in which order we process the cycles. To this end, we define an auxiliary bipartite graph $\mathcal{H}$ whose first (second) part has a vertex for each connected component in H_x (H_y), and there is an edge between two vertices of $\mathcal{H}$ if and only if the corresponding components in H_x and H_y share at least one vertex; see Fig. 6.1. We will first establish a structural property of $\mathcal{H}$. In the following, we will always refer to the corresponding vertex of a component c in H_x or in H_Y that belongs to $\mathcal{H}$ as v_c.

Property 5. *The auxiliary graph $\mathcal{H}$ is connected.*

Proof. We suppose for a contradiction that $\mathcal{H}$ is not connected. Then, there exist two vertices v_c and $v_{c'}$ in $\mathcal{H}$ which belong to different components of $\mathcal{H}$. We assume that $c \in H_x$. Let v_1 and v_k be two vertices of G which belong to c and c', respectively. Since G is connected, there exists a path $P = (v_1, \ldots, v_k)$ in G. By the matching property, we have that no two consecutive edges of P can belong to the same matching. Let (v_i, v_{i+1}) be the first edge when traversing P starting at v_1 which belongs to M_3. Such an edge exists, as otherwise we would have that $c = c'$. Hence, v_i belongs to c and to some other component $c^* \in H_y$, but then v_c and v_{c^*} are connected in $\mathcal{H}$ by construction. Repeating this argument until v_k is reached yields a path in $\mathcal{H}$ from v_c to $v_{c'}$, a contradiction. □

We will now use a traversal of $\mathcal{H}$ in order to construct two total orders $\prec_x$ and $\prec_y$ of the vertices of G which will be used to derive the final x- and y-coordinates of them. Recall that since the edges of M_1 (M_3) are drawn vertical (horizontal), we require that the endpoints of an edge of M_1 (M_3) are consecutive in $\prec_x$ ($\prec_y$). If we



(a) G (b) $c_1, \ldots, c_5$

(c) $\mathcal{H}$

FIGURE 6.1: (a) A non-planar, non-Hamiltonian, 3-edge-colorable degree-3 example graph G. The matching M_1 is drawn with dark blue, M_2 with red and M_3 with light blue. (b) The components c_1, c_4 and c_5 of the subgraph H_x induced by $M_1 \cup M_2$ (shaded in green) and the components c_2 and c_3 of H_y induced by $M_2 \cup M_3$ (shaded in orange). (c) The auxiliary graph $\mathcal{H}$ in which the components of H_x and H_y that share at least one vertex are connected by an edge. In this example, one valid BFS traversal of the components of $\mathcal{H}$ would be c_1, c_2, c_3, c_4, c_5.

encounter a vertex of $\mathcal{H}$ whose corresponding component belongs to H_x (H_y) during our traversal, we will append all its vertices to $\prec_x$ ($\prec_y$), where the exact ordering of these vertices will be specified shortly. In a broad sense, the traversal is a simple BFS traversal of $\mathcal{H}$. To select the first vertex of the traversal of $\mathcal{H}$, we choose an (almost) arbitrary vertex u of G. As observed earlier, every vertex, thus in particular u, belongs to two components c and c' of H_x and H_Y, respectively. If c is a path, we require that u is the endpoint of c; in the case of c being a cycle we impose no constraints. We call u the *origin vertex* of G. We start our BFS traversal of $\mathcal{H}$ at v_c and then move to $v_{c'}$ in the second step (clearly, these vertices are connected in $\mathcal{H}$ as the corresponding components both contain the origin vertex). From now on, we continue the BFS traversal of the remaining vertices of $\mathcal{H}$ without any additional restrictions. This specific choice of the first step will be of importance for a structural property of an edge which belongs to M_2. Now that we have described a valid order in which the components of H_x and H_y are processed, we turn to the ordering of the vertices of a specific component to guarantee Property 6). Let c be the component which corresponds to the currently visited vertex in the traversal of $\mathcal{H}$. By bipartiteness, it follows that no other component of H_x shares a vertex with c if c belongs to H_x. Hence, it follows that no vertex of c is already part of H_x (an analogous statements clearly holds for H_y). If c is a path, then we append the vertices of c to $\prec_x$ ($\prec_y$) in the order defined by a walk from one endpoint of c to the other. For the special case that c is a path and the first component in the BFS traversal of $\mathcal{H}$, one of

the endvertices of c is in fact the origin vertex by definition, which we then choose as the start of our walk. Hence, in the remainder we have to discuss how to handle the case where c is a cycle. The main idea is to again append the vertices to $\prec_x$ or $\prec_y$ in the order derived by a walk along c such that the closing edge of c necessarily belongs to M_2. However, the choice of the starting vertex for the walk is a bit more involved. Suppose first that $c \in H_x$. If c is the first component of the traversal of $\mathcal{H}$, then we choose the origin vertex of G as the starting vertex and define the walk such that the edge of M_1 incident to the origin vertex is the first one. Otherwise, if c is not the first component, some vertices of c were already placed in $\prec_y$ beforehand (as the previous "level" of the BFS traversal considered components of H_y and at least one of them shares a vertex with c (which is in fact the parent of c). Let v be the first vertex of c in $\prec_y$. We start our walk at v and again follow the edge of M_1 that is incident to v. We remark here the important property that v is the first vertex of c in both $\prec_x$ and $\prec_y$ by construction.

Suppose now that $c \in H_y$. Then c cannot be the first component of the traversal of $\mathcal{H}$. Similar to the previous case, let v be the first vertex of c in $\prec_x$, which is again well defined as at least one vertex of c is already part of $\prec_y$ (i.e., the one that is shared between c and its parent w.r.t. the traversal). We start our walk of c at v and follow the edge of M_3 that is incident to v which uniquely defines our walk of c. However, contrary to the previous case, we will append the vertices of c to $\prec_y$ in the **inverse** order that they appear in this walk. Observe that v is the first vertex of c in $\prec_x$, but the last vertex of c in $\prec_y$. Again, as c has even length, we have that the closing edge belongs to M_2, see Fig. 6.2 for an illustration. Note that by construction, the closing edge uv of any cyclic component c belongs to M_2, and since one endpoint of uv was chosen such that it is also contained inside the parent component, it follows that v also belongs to it by definition. Moreover, we have the following property.

Property 6. *The endvertices of any edge in M_1 (M_3) are consecutive in $\prec_x$ ($\prec_y$). The endvertices of any edge in M_2 are consecutive in $\prec_x$ ($\prec_y$) unless this edge is a closing edge in a component of H_x (of H_y).*

Computing the vertex coordinates. With the total orders $\prec_x$ and $\prec_y$ at hand, we can now compute the explicit x- and y-coordinates of the vertices respectively. Since our drawing is straight-line, this fully describes the final drawing Γ. We will first describe how to assign the x-coordinates. Initially, we assign the first vertex of $\prec_x$ the x-coordinate 1. Now, we iterate through $\prec_x$. let v be the next vertex of the iteration and let u be its predecessor in $\prec_x$. Assume that the x-coordinate of u is i. If $uv \in M_1$, then v gets the same x-coordinate i as u. Otherwise, $uv \in M_2$ and v gets x-coordinate $i+1$. Similarly, we iterate through $\prec_y$, set the y-coordinate of the first vertex to 1 and if v is the next vertex in the iteration and its predecessor u had y-coordinate i, vertex v is assigned y-coordinate i if $uv \in M_3$, and y-coordinate $i+1$ otherwise. Observe that by construction, no two vertices u and v are placed on the same positions, as otherwise they would necessarily be connected by an M_1 edge and by an M_3 edge, which is impossible as the input graph is simple.

It remains to show that the computed vertex coordinates induce a straight-line RAC drawing Γ of G with the possible exception of exactly one edge e^*. Edge e^*, if it exists, is the edge of M_2 incident to the origin vertex of G. Observe that by construction, e^* is the closing edge of two components, and therefore its endpoints are neither consecutive in $\prec_x$ nor in $\prec_y$ by Property 6. We will describe how to incorporate e^* into the final drawing in the end, hence for now we consider graph $G^* = G \setminus e^*$ and compute its drawing Γ^*.

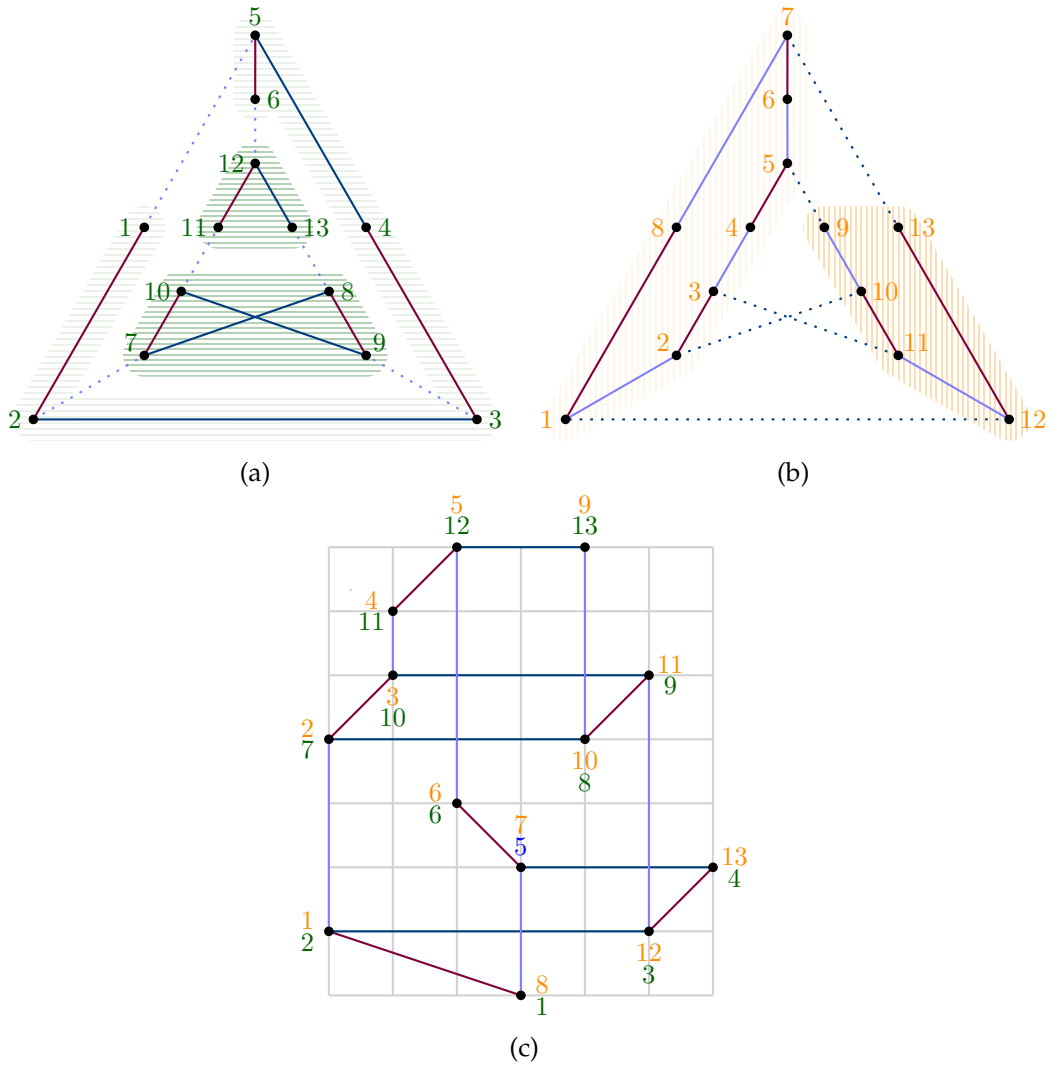


FIGURE 6.2: The total orders (a) $\prec_x$ for H_x that consists of the dark blue and red edges, and (b) $\prec_y$ for H_y that consists of the light blue and red edges. The final drawing of G is shown in (c).

Lemma 17. *Let e be an edge of G^*. Then, e is drawn vertically in Γ^* if $e \in M_1$; horizontally in Γ^* if $e \in M_3$ and crossing-free in Γ^* if $e \in M_2$.*

Proof. If the edge e belongs to M_1 or M_3, the statement directly follows from Property 6 and the computed vertex coordinates. Hence assume that $e = uv \in M_2$ and let $c_x \in H_x$ and $c_y \in H_y$ be the two components that contain e. Suppose for a contradiction that there exist an edge $e' = u'v'$ which crosses e. Let us first establish some preliminary observations about e'. If $e' \in M_1$, then both u' and v' belong to the same component $c'_x \in H_x$. If c_x and c'_x are different then by construction the vertices of c_x and c'_x span different x-intervals, therefore they cannot possible intersect, thus e' belongs to c_x. Similarly, if $e' \in M_1$, then e' also belongs to c_y. Finally, by an analogous argument, if $e' \in M_2$, then e' belongs to the same component in both H_x and H_y, in particular, e' belongs to c_x and c_y. In the following case analysis, we will distinguish base on the type of e and e'.

1. Edge e is a closing edge for neither c_y nor c_x.

   It follows by Property 6 that u and v are consecutive in both $\prec_x$ and $\prec_y$. In particular, by construction, their x- and y-coordinates differ by exactly one.

Since vertices are placed on integer coordinates, no vertical or horizontal edge can intersect e, hence no edge of M_1 or M_3. Hence, e' necessarily belongs to M_2. By construction, the x- and y-coordinates of the vertices in c_y and c_x are non-decreasing along the walk that defines their order in $\prec_x$ and $\prec_y$. Hence, e' cannot cross e unless it is a closing edge in either c_x or c_y (or both). This case will be covered later by swapping e and e'.

2. Edge e is a closing edge for c_x but not for c_y.

   Recall that since we argue about G^*, component c_x cannot be the first component of the BFS traversal of $\mathcal{H}$ as otherwise $e = e^*$. Since e is not closing in c_y, we have that u and v are consecutive in $\prec_y$ by Property 6. W.l.o.g. assume that u directly precedes v in $\prec_y$, which then implies that u is the first vertex of c_x in $\prec_y$ by our construction rule, as when we are considering c_x, we start our walk at the first vertex of c_x in $\prec_y$. As the y-coordinate of u and v differs by exactly one, it follows that e' cannot belong to M_3. If e' belongs to M_1, then one of u' or v', say u' has y-coordinate smaller or equal than u. But since u and v are consecutive in $\prec_y$, it follows that u' precedes u in $\prec_y$, a contradiction to our choice of u as u' also belongs to c_x.

3. Edge e is a closing edge for c_y but not for c_x.

   The argumentation for this is case is symmetric to the previous one.

4. Edge e is a closing edge for both c_y and c_x.

   In the following, we will establish that only the edge of M_2 incident to the origin vertex potentially has this property, which means this case cannot occur in G^*. As we are considering G^*, neither c_x nor c_y can be the first component in the BFS traversal of $\mathcal{H}$. Recall that by definition, the vertices v_{c_y} and v_{c_x} are adjacent in $\mathcal{H}$ as they share the endpoints of e. Assume that c_x is visited before c_y in the BFS traversal; the other case is symmetric. While considering c_x, we start our walk from the vertex u which is by construction the first vertex of c_x in $\prec_y$, which means u also belongs to a component c'_y of H_y. As c_x is a cycle (clearly the considered case cannot occur otherwise), u is incident to an edge in M_2 which also belongs to c'_y. The edge of M_2 incident to c_x is the closing edge of c_x, which also belongs to $c'_y \neq c_y$ by assumption. Since the components of H_x and H_y are disjoint, it follows that the case cannot occur in G^* which concludes the proof.

□

The last case of the previous lemma shows that the edge e^*, if it exists, is the only edge that is closing in both components which contain it.

Corollary 4. *There is at most one edge in M_2 that is a closing edge for two components.*

We will now describe how to alter our drawing Γ^* such that we can add the edge e^* and such that all the remaining edges still admit a RAC drawing. The final drawing will be denoted by Γ. Clearly, if e^* is missing, then $\Gamma = \Gamma^*$ and we are done. Otherwise, denote by u and v the endvertices of e^* such that u is the origin vertex of G. As $e^* \in M_2$ and by our construction rule, both u and v are contained in the first two components of the BFS traversal of $\mathcal{H}$ which we denote by c and c'. Since u is the first vertex of $\prec_x$, its x-coordinate is 1, i.e., u is the leftmost vertex of Γ^*. By construction, we further have that u is the last vertex of c' in $\prec_y$ (as we use the inverse ordering) and since uv is also a closing edge of c', it follows that v is

the first vertex of c' in $\prec_y$ and hence the first vertex of G in $\prec_y$ (as c' is exactly the second component in the BFS traversal after c). In particular, this implies that v has y-coordinate 1, i.e., it is the bottom most vertex of G^*.

We will move u to the left and v to the bottom in order to add the edge e^*. In particular, by moving u by n units to the left and v by n units to the bottom, we have that the segment corresponding to e^* does not intersect the quadrant $\mathbb{R}^2_+$, which contains all edges of G that are not incident to u or v. Clearly, e^* also does not cross any edge incident to u or v. It remains to show that the incident edges to u or v are still valid. First observe that e^* is the only edge of M_2 that is incident to u or v by the matching property. The edge of M_3 incident to u and the edge of M_1 incident to v are simply elongated and hence stay horizontal and vertical, respectively. Finally, we consider the M_1-edge e' incident to u, the case of the M_3-edge incident to v can be argued symmetrically. We claim that e' was crossing-free in Γ^*. To see this, we observe that an edge crossing e' (which is a vertical edge) is incident to a vertex that has a strictly smaller x-coordinate than u, which is a contradiction. Thus, moving u to the left does not introduce any new crossings. Together with Lemma 17 we obtain that Γ is a RAC drawing of G.

It remains to discuss the time-complexity and the area requirement to compute our drawing Γ. Recall that we assume that the 3-coloring is given as part of the input. Then we can construct the components of H_x and H_y in linear time. As every vertex is contained in exactly two components in $H_x \cup H_y$, it follows that $\mathcal{H}$ has $\mathcal{O}(n)$ many edges, whence the BFS traversal requires $\mathcal{O}(n)$ time. Choosing the origin vertex can trivially be performed in linear time. In order to construct $\prec_x$ and $\prec_y$, we have to traverse every edge of G at most twice, hence this step also takes $\mathcal{O}(n)$ time. Finally, with $\prec_x$ and $\prec_y$ at hand, we can compute the vertex-coordinates by iterating through $\prec_x$ and $\prec_y$ once in $\mathcal{O}(n)$ time. The post-processing (i.e., potentially adding the edge e^*) can be done in constant time. Thus, the total time-complexity to compute Γ is $\mathcal{O}(n)$.

Concerning the area, we observe that the initial x and y coordinates for the vertices range between 1 and n. In order to add the special edge e^* which is incident to the origin vertex u and vertex v, we have to move u n-units to the left and v n-units to the bottom, thus the resulting drawing area is at most $2n \times 2n$.

6.3 1-bend RAC drawings of degree-4 graphs

We now turn our attention to graphs which admit 1-bend RAC drawings. In particular, we will prove that every degree-4 graph admits such drawing. To this end, we provide two different algorithms to produce such a drawing. The first one is based on the result derived in the previous section. The rough idea is to split every vertex of the original graph into two to obtain a 3-colorable degree-3 graph. We then apply Theorem 10 to this graph such that, in the final drawing, we will place the bend-points of the edges sufficiently close to the computed position of the split-vertex. The other approach is an extension of the result in [13], where the authors show that every degree-3 graph admits a 1-bend RAC drawing. The benefit of the latter approach is that we can guarantee a linearly sized set of edges which are drawn straight-line, whereas in the first construction every edge requires a bend to be drawn.

Theorem 11. *Given a degree-4 graph G with n vertices, it is possible to compute in $O(n)$ time a 1-bend RAC drawing of G with $O(n^2)$ area.*

Proof. By Theorem 9, we can augment G into a directed 4-regular multigraph G' with 2-factors F_1 and F_2. Recall that F_1 and F_2 are by construction edge disjoint. We will now compute a *split-graph* G_s of G' as follows. By definition, every vertex u of G' has exactly one incoming and one outgoing edge in F_1 and F_2, each; denote these by $(a_1, u), (u, b_1) \in F_1$ and $(a_2, u), (u, b_2) \in F_2$, respectively. We add two vertices u_s and u_t to G_s such that u_s is incident to the two incoming edges of u, while u_t is incident to the two outgoing edges of u (with respect to F_1 and F_2). In particular, we have $(a_1, u_s), (a_2, u_s), (u_s, b_1)$ and (u_s, b_2). Finally, we will add the edge (u_s, u_t) to G_s, which is called *split-edge*. Clearly, G_S is 3-regular by construction. To see that G_s is also 3-edge colorable, we observe that every vertex of G_s is incident to exactly one edge of F_1, one edge of F_2, and one split-edge. Hence, coloring the edges of F_1, the edges of F_2 and the set of split-edges in three different colors provides a valid 3-edge coloring of G_s. Therefore, all the preconditions of Theorem 10 are met and we can compute a RAC drawing Γ_s of G_s. In particular, we choose M_2 to consist of exactly the set of split-edges in G_s. It remains to transform Γ_s into a 1-bend drawing of G'. To do so, we first place vertex u of G' at the computed position of u_s in Γ_s. We will draw every outgoing edge (u, x) of u in $F_1 \cup F_2$ as a polyline with a bend that is placed sufficiently close to the position of u_t in Γ_s. Hence, the two segments of (u, x) are close to (u_s, u_t) and (u_t, x) in Γ_s, respectively. Since any edge is outgoing for exactly one of its endvertices, it follows that every edge has exactly one bend. In the following, we will describe the exact position of such a bend point in detail. Let us fix vertex u and describe its outgoing edges. Recall that the set of split-edges belongs to M_2, hence it corresponds either to the diagonal of a 1×1 grid box, or it is a closing edge. The outgoing edges of u_t belong to M_1 or M_3 and are therefore drawn vertical or horizontal in Γ_s.

Assume first that the split-edge (u_s, u_t) of u is the diagonal of a 1×1 grid box, see Fig. 6.3a for an illustration. If the edge (u_t, x) belongs to M_1, i.e., it is a vertical segment in Γ_s, then we place the bend either half a unit above u_t if x is above u_t or half a unit below u_t otherwise. Symmetrically, if (u_t, x) belongs to M_3, i.e., it is a horizontal segment in Γ, then we place the bend either to the left or to the right of u_t (depending on the relative position of u_t and x in x-direction).

Assume now that the split-edge of u is a closing edge in exactly one component of H_x or H_y. We consider here the case where it is the closing edge of a cycle $c \in H_x$, the other case is symmetric. Recall that a closing edge of a cycle in H_x necessarily spans the whole x-interval of c. Again, we distinguish based on the type of the edge (u_t, x). If $(u_t, x) \in M_1$, then (u_t, x) corresponds to a vertical segment in Γ_s by construction, in which case we place the bend-point exactly at the position of u_t in Γ_s. The important observation is that x belongs to the same component C of M_1 as u_s, and since u_s is incident to the closing edge of C, we have by construction that u_s is the first vertex of C in $\prec_y$, hence x is necessarily above u_t. sIf $(u_t, x) \in M_3$, i.e., (u_t, x) corresponds to a horizontal segment in Γ_s, we place the bend-point half a unit to the left of the position of u_t in Γ_s if $x \prec_x u_t$; otherwise we place it half a unit to the right, see Fig. 6.3b for an illustration.

Finally, assume that the split-edge of u corresponds to a closing edge e of a cycle in H_x and a cycle in H_y. Recall that this edge is unique by Corollary 4. As discussed at the end of the previous section Section 6.2, one of u_s and u_t is the leftmost vertex, while the other one is the bottommost vertex in Γ_s. W.l.o.g. assume that u_s is the leftmost vertex in Γ_s. For this special case, we have to slightly alter the previous scheme. Namely, let (u_t, x) and (u_t, y) be the two edges of M_1 and M_3, respectively, that are incident to u_t in G_s. By construction, if we elongate the segment of e in G_s to a line, we have that all the vertices of Γ_s lie in the half-plane above e. We will place

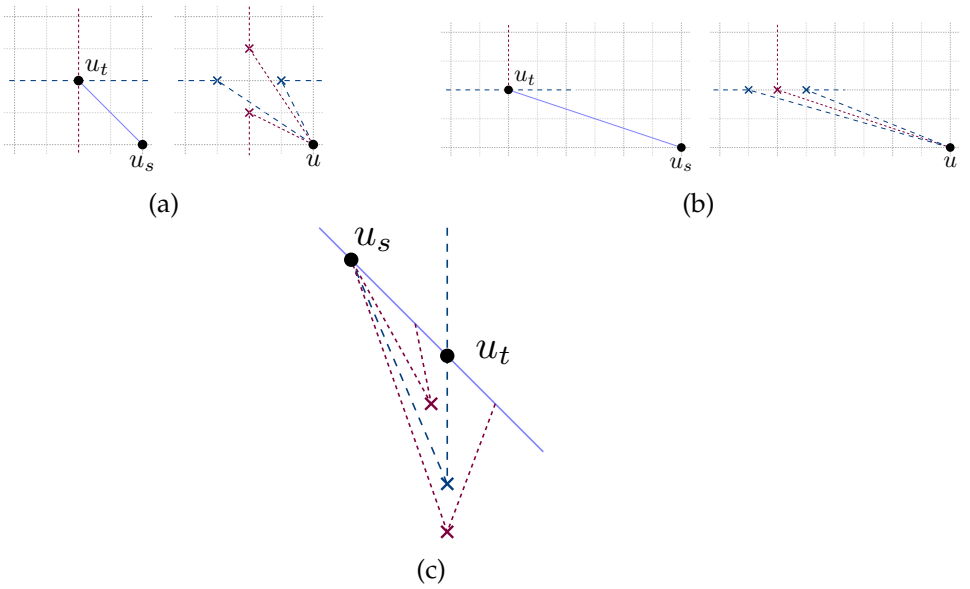


FIGURE 6.3: Illustration on how to place the bends in the proof of Theorem 11. To merge the vertices u_s and u_t of a vertex u in G', u is placed at the position of u_s. The bends of the outgoing edges at u are placed close to the position of u_t in the drawing depending on their orientation.

the bend-point b_x of (u_t, x) sufficiently below u_t, i.e., it has the same x-coordinate as u_t and y-coordinate $-2n$. If $y \prec_x u_t$, then we will place the bend-point b_y of (u_t, y) a unit to the left and two units above b_x, otherwise, we place b_y a unit below b_x, see Fig. 6.3c for an illustration.. Clearly, (u, x) and (u, y) drawn by bending at b_x and b_y do not cross each other. Further, their first parts are fully contained in the half-plane below e, hence they do not intersect any other edge. Finally, the second part of (u_t, x) remains vertical, while the second part of (u_t, y) is crossing free since (u_t, y) was necessarily crossing free in Γ_s to not obtain a contradiction to u_t being the bottommost vertex of Γ_s.

Regarding the time complexity, we observe that we can apply Theorem 9 and the split-operation in $\mathcal{O}(n)$ time. The split operation immediately yields a valid 3-coloring of the edges, hence we can apply the algorithm of Theorem 10 to obtain Γ_s in $\mathcal{O}(n)$ time. Finally, contracting the edges can clearly be done in $\mathcal{O}(n)$ time, as it requires a constant number of operations per edge. For the area, we observe that in order to place the bends, we have to introduce new grid-points, but we at most double the number of points in any dimension and for the special edge, we double the size of the grid, hence we still maintain the asymptotic quadratic area guaranteed by Theorem 10. □

We will now present the second algorithm that produces 1-bend RAC drawings for every degree-4 graph.

Theorem 12. *Given a degree-4 graph G with n vertices and m edges, it is possible to compute in $O(n)$ time a 1-bend RAC drawing of G with $O(n^2)$ area where at least $\frac{m}{8}$ edges are drawn as straight-line segments.*

Proof. By applying Theorem 9, we augment G to a directed 4-regular multigraph G' and obtain the two edge disjoint 2-factors F_1 and F_2. Throughout the remainder of the section, we will in fact assume that G and G' are equivalent, as the drawing of G

can easily be inherited from the computed drawing of G'. Moreover, we will also assume that G is simple - while this cannot be guaranteed when applying Theorem 9 to the input graph, we can simply omit a copy for every pair of multiple edges, which clearly simplifies the construction. The main idea of the algorithm is to compute a total order $\prec$ of the vertices of G based on F_1. Afterwards, we will place vertex v with position i in $\prec$ at position (i, i) in the final drawing Γ such that the edges of F_1 (minus the closing edges) can be drawn on the diagonal using straight-line segments. Each edge of F_2 will be drawn as a polyline which consists of exactly one horizontal and one vertical segment. To be more precise, for a directed edge $(u, v) \in F_2$, we have a horizontal segment incident to u and a vertical segment incident to v. If $u \prec v$, then (u, v) will be drawn below the diagonal, i.e., the horizontal segment at u uses the E-port of u and the vertical segment incident to v uses the S-port at v. Conversely, if $v \prec u$, then (u, v) will be drawn above the diagonal, i.e., the horizontal segment at u uses the W-port of u and the vertical segment at v uses the N-port of v. Since every vertex has indegree and outdegree one in F_2 by construction, this scheme will not use the same port twice for any vertex of G. Hence, it remains to incorporate the closing edges of F_1 into the drawing.

Remark 1. *If G is Hamiltonian, we can already stop the description of our algorithm at this point, since we can choose F_1 such that it coincides with the Hamiltonian cycle, in which case F_1 contains exactly one closing edge between the first and the last vertex of $\prec$. By construction, they are the bottommost and the rightmost vertex of our drawing and thus they can be joined by two crossing-free segments.*

Let $\Gamma_{\overline{D}}$ be the induced drawing of $G \setminus D$, where D is the set of closing edges. The following lemma follows directly from the routing-scheme of the edges of F_2.

Lemma 18. *Any vertex v in $\Gamma_{\overline{D}}$ has two free orthogonal ports. Moreover, the free orthogonal ports are opposite to the used ones.*

Proof. By construction, we only have to consider edges of F_2. Suppose first that v uses the W- and E-port. A used E-port implies the existence of an edge (v, u_i) (with $v \prec u_i$), while a used W-port implies the existence of an edge (v, u_j) (with $u_j \prec v$), but then v has two outgoing edges in F_2, a contradiction. Equivalently, a used N-port and a used S-port each imply the existence of an incoming edge, which is a contradiction to the definition of F_2. □

We will now define $\prec$ precisely such that it satisfies some desired properties. In particular, we say an orthogonal port at vertex u is *good* with respect to an edge (u, v), if the port can be used to "draw" the edge. More formally, if $u \prec v$, then the E-port and the N-port of u are good. Otherwise, if $v \prec u$, then the W-port and the S-port of u are good.

Lemma 19. *There exists a total order $\prec$ on the vertices of G such that, for any cycle c of F_1, one of the two endpoints incident to the closing edge of F_1 has a free good port (with respect ot the closing edge).*

Proof. In order to achieve the desired total order, we will fix an initial ordering of the cycles $C_1, \ldots, C_k$ such that all vertices of C_i appear before all vertices of C_j in $\prec$ if $i < j$. Clearly this property is maintained after a local operation which involves a set of vertices that all belong to the same cycle. In particular, a cyclic rotation of the vertices of a cycle C w.r.t to $\prec$. is such a local operation. In the following, we will show that we can prove the lemma, i.e. construct a desired total order $\prec$ by only applying cyclic rotations independently for each cycle. Hence, assume that cycle C

does not satisfy this property, i.e., neither of the endpoints of the closing edge (with respect to the current $\prec$) has a free good port. First observe that not all vertices of C can be incident to only internal edges in F_2, as otherwise G would be disconnected. Hence, let v be a vertex that has an external edge. If this external edge is a backward edge, then rotating C such that v is the first vertex of C in $\prec$ implies that either the N-port or the E-port of v is free (as v has degree two in F_2), which are both good ports. Similarly, if it is a forward edge, rotating C such that v is the last vertex of C in $\prec$ implies a free S-port or a free W-port. □

We will now describe how to add the closing edges D to $\Gamma_{\overline{D}}$ by leveraging Lemma 18 and Lemma 19, where the latter implies that we have at least one good port free for any closing edge. Let C be a cycle of F_1. W.l.o.g. assume that the last vertex of C in $\prec$ has a free S-port. The other cases are symmetrical. Let $v_1, v_2, \ldots, v_k$ be the ordering of the vertices of C induced by $\prec$. Recall that by construction, the position of vertex v_i is $(i + x, i + x)$, where x is the total number of vertices which are contained in cycles that precede C in $\prec_x$. For the following argumentation, we will omit this offset x when talking about the position of the vertices of C for clarity reasons, hence we can assume that v_i is at position (i, i). We will now consider the port assignment of the edges incident to v_1 which gives rise to the following cases.

a) v_1 has a free E-port.

   Since v_k has a free S-port by assumption, we simply add the closing edge (v_1, v_k) to $\Gamma_{\overline{D}}$ drawn as a polyline with a horizontal segment incident to v_1 and a vertical segment incident to v_k. Hence, for the remainder we can assume that the E-port of v_1 is not free.

b) The E-port and the S-port of v_1 are not free.

   Observe that since v_1 is the first vertex of C in $\prec$, it follows that the edge using the S-port of v_1 is a backward edge. Consider now the edge (v_1, v) that incident to v_1 which is using the E-port of v_1. If (v_1, v) is a forward edge, then we have that $v_1 \prec v$ and $v \notin C$ and we move v_1 to position $(k, 2 - \epsilon)$. This allows us to draw the edge (v_1, v_k) using only one vertical segment that uses the N-port at v_1, which is free by Lemma 18. By assumption, since (v_1, v) is a forward edge, we still have that the x-coordinate of v is larger than the one of v_1, thus the edge (v_1, v) can still be drawn such that its segment incident to v_1 uses the E-port of v_1. Similarly, this holds for the backward edge whose segment incident to v_1 uses the S-port. Finally, we have to consider the edge $(v_1, v_2) \in F_1$. We will draw it as a polyline with a horizontal segment at v_1 which uses the W-port of v_1 such that the bend occurs sufficiently close to v_2; hence the segment incident to v_2 is crossing free and will not be assigned an orthogonal port, see Fig. 6.4a for an illustration. Otherwise, (v_1, v) is an internal edge. In particular, v corresponds to a vertex $v_i \in C$. We then place v_1 at $(i, 2 - \epsilon)$. This allows us to draw the edge (v_1, v) using a single vertical segment that uses the N-port at v_1. Further, since $v_i \prec v_k$, we have that v_k is to the left of the new position of v_1 - thus we can draw the edge (v_1, v_k) as a polyline with a horizontal segment incident to v_1 that uses the E-port and a vertical segment that uses the free S-port at v_k. The edge (v_1, v_2) as well as the backward edge are drawn analogous to the previous case, see Fig. 6.4b for an illustration.

c) The E-port and the N-port of v_1 are not free and at least one of the edges which occupy these ports is external.

We first consider the case where both edges whose incident segments to v_1 use the E- and the N-port are forward edges. In this case, we place v_1 at $(n+\epsilon, 2-\epsilon)$. This ensures that we can draw the edge (v_1, v_k) as a polyline which consists of a non-orthogonal segment incident to v_1 and a vertical segment incident to the S-port of v_k, such that its bend point is placed at $(n, 2)$. The edge (v_1, v_2) is drawn analogous to the first case and both forward edges keep the same port at v_1, see Fig. 6.4c for an illustration.

Consider now the case where the edge that is using the E-port at v_1 is a forward edge, while the edge that uses the N-port at v is internal. Let v_i be the other endpoint of this internal edge. In particular, since (v, v_i) uses the N-port at v and since $v \prec v_i$, it follows that (v, v_i) uses the W-port at v_i. By Lemma 18 we obtain that the E-port at v_i is free. We use this by placing v_1 at $(k-\epsilon, 2-\epsilon)$ which allows us to add (v_1, v_i) using the N-port of v_1 and the E-port of v_i. Similar to the previous case, we will draw the edge (v_1, v_k) as a polyline such that the segment incident to v_1 is not orthogonal, while the segment incident to v_k uses the S-port at v_k. The forward edge which uses the E-port of v_1 will remain unchanged, while the edge (v_1, v_2) will be drawn as in the previous cases, see Fig. 6.4d.

For the last case, assume that the edge that uses the N-port at v_1 is external (i.e., it is a forward edge), while the edge using the E-port at v_1 is internal. Again, we denote by v_i the other endpoint of the internal edge. By Lemma 18, either the W-port or the E-port of v_i is free, w.l.o.g. we assume that the W-port of v_i is free, the other case is symmetric. We place v_1 at $(i-\epsilon, 2-\epsilon)$. Moreover, we will redraw the edge (v_{i-1}, v_i) as follows. Recall that initially, since (v_{i-1}, v_i) is a non-closing edge of F_1, it is drawn as a straight-line segment between v_{i-1} and v_i and lies on the diagonal. We draw (v_{i-1}, v_i) as a polyline which, starting at v_{i-1}, uses no orthogonal port at v_{i-1}, bends at point $(i-1+\epsilon, i)$ such that the segment incident to v_i uses the W-port of v_i which is free by assumption, see Fig. 6.4e for an illustration. In particular, since v_1 is placed at x-coordinate $i-\epsilon$, the forward edge which uses the N-port at v_1 can intersect the edge (v_{i-1}, v_i). The remaining edges are then drawn similar to the previous cases. In Fig. 6.4f, we show the symmetric case where the E-port of v_i is free.

d) Both edges of F_2 that are incident to v_1 are internal.

Denote by v_t (v_r) the endpoint of the edge that is using the N-port (the E-port) of v_1, respectively. Observe that in the initial drawing, the edge (v_1, v_r) uses the S-port at v_r by construction. Moreover, we have by Lemma 18 that the E-port of v_t is free (as its W-port is used by (v_1, v_t)). Assume first that $t < r$. We place v_1 at $(r-\epsilon, 2-\epsilon)$ and draw the edge (v_1, v_r) such that its segment incident to v_1 is not orthogonal and the segment incident to v_r still uses the S-port. Since $t < r$, it follows that the new position of v_1 is to the left of v_t. Hence, we can draw (v_1, v_t) such that it uses the N-port at v_1 and the E-port at v_t, where the former is free as it was previously occupied by (v_1, v_r) and the latter is free as shown earlier. The edge (v_1, v_k) will use the E-port at v_1 and the S-port at v_k. Finally, the edge (v_1, v_2) will be drawn as usual, see Fig. 6.4g.

Assume now that $r < t$. By Lemma 18, either the W-port or the E-port of v_r is free, w.l.o.g. we assume that the E-port is free. Then, we place v_1 at $(r+\epsilon, 2-\epsilon)$. Again, we redraw the edge (v_r, v_{r+1}) such that the edge (v_1, v_t)

can cross the spine; see Figs. 6.4h and 6.4i. The other edges are analogous to the previous cases which concludes the construction.

To see that the constructed drawing is indeed 1-bend RAC, we observe that (i) every edge consists of at most two segments, (ii) every edge-segment that is neither horizontal nor vertical either lies on the diagonal of the drawing which is not crossed (at this part of the drawing), or it is constructed such that it spans ϵ in one direction and exactly 1 in the other direction which both define empty closed intervals in Γ. Before we bound the number of edges that require a bend, we first consider the area requirement and the time complexity to construct Γ. Applying Theorem 9 to obtain the 2-factors can be done in $\mathcal{O}(n)$ time, which gives us an initial total ordering of the vertices $\prec_{init}$. Defining a feasible total order of the vertices for every cycle $\prec$ given $\prec_{init}$ can be done in $(|C|)$ time for each cycle individually and hence in $\mathcal{O}(n)$ time in total. The potential displacement of the first vertex of a cycle in $\prec_y$ requires a constant number of operations, hence in total $\mathcal{O}(n)$ time. For the area, we observe that the initial positioning on the diagonal takes $n \times n$ area (including the bends). By setting ϵ to $\frac{1}{3}$, it is sufficient to scale the final drawing by a factor of 3 in order to guarantee grid coordinates for vertices and bends, hence we obtain a quadratic drawing area.

In order to bound the number of edges that require a bend, let $C = (v_1, v_2, \ldots, v_n)$ be a cycle of F_1. By construction, all edges of F_1 consist of one segment with the exception of possibly (i) the first edge of C, (ii) the last (i.e., the closing edge) of C and (iii) one intermediate edge. In the following, we will refer to the original edges of the input graph as *real edges*, while the ones that were introduced by applying Theorem 9 are called *fake edges*. Among the two 2-factors of G, choose F_1 such that it contains the most real edges. Let C be a cycle of F_1. Denote by $r(C)$ the number of real edges of C and by $s(C)$ the number of edges of C which consist of only one segment in Γ. Clearly, if C contains a fake edge, we can rotate C such that the closing edge corresponds to the fake edge, in which case $s(C) = r(C)$ follows, as all vertices of C can be placed on the diagonal. Assume now that C does not contain a fake edge. Clearly, we have $|C| \geq 3$, as we assume the initial graph to be simple. First assume that $|C| = 3$. Since C cannot contain internal edges, it follows that all edges of F_2 which are incident to vertices of C are external. As the vertices of C form a complete graph, if there is a vertex $v_1 \in C$ incident to two forward (backward) edges, while one of its neighbors $v_2 \in C$ is incident to at most one forward (backward) edge, then we define the closing edge of C to be (v_1, v_2) and define $\prec$ such that v_1 is the last (first) vertex of C in $\prec$, while v_2 is the first (last), which allows us to add the closing edge without moving any vertex. Hence we now assume that all the vertices of C have two forward edges (the case where all vertices have two backward edges is symmetric). Consider an arbitrary ordering of the vertices v_1, v_2 and v_3 in C, i.e., $v_1 \prec v_2 \prec v_3$. Since all vertices have two forward edges, it follows that the S-port and the W-port of all vertices is free. In particular, the S-port of v_2 is free. We will draw the closing edge (v_1, v_3) such that it uses the S-port of v_3 and such that it uses a non-orthogonal segment at v_1. This segment spans an x-interval that contains exactly one vertex, namely v_2, in its interior. Since the S-port of v_2 is free and since v_2 is not incident to any closing edge of F_1, it follows that the segment of (v_1, v_3) incident to v_1 is crossing free. Hence, we conclude that in both cases, $s(C) = r(C) - 1$. For $|C| \geq 4$, at most $|C| - 3$ edges that belong to F_1 consist of two segments, hence we obtain $s(C) \geq \frac{r(C)}{4}$. By the choice of F_1, we have that $r(F_1) \geq \frac{m}{2}$, where m is the number of edges of the original graph. Clearly, $r(F_1) = \sum_i r(C_i)$, as the cycles are edge-disjoint by construction. If C is a cycle with a fake edge, we have

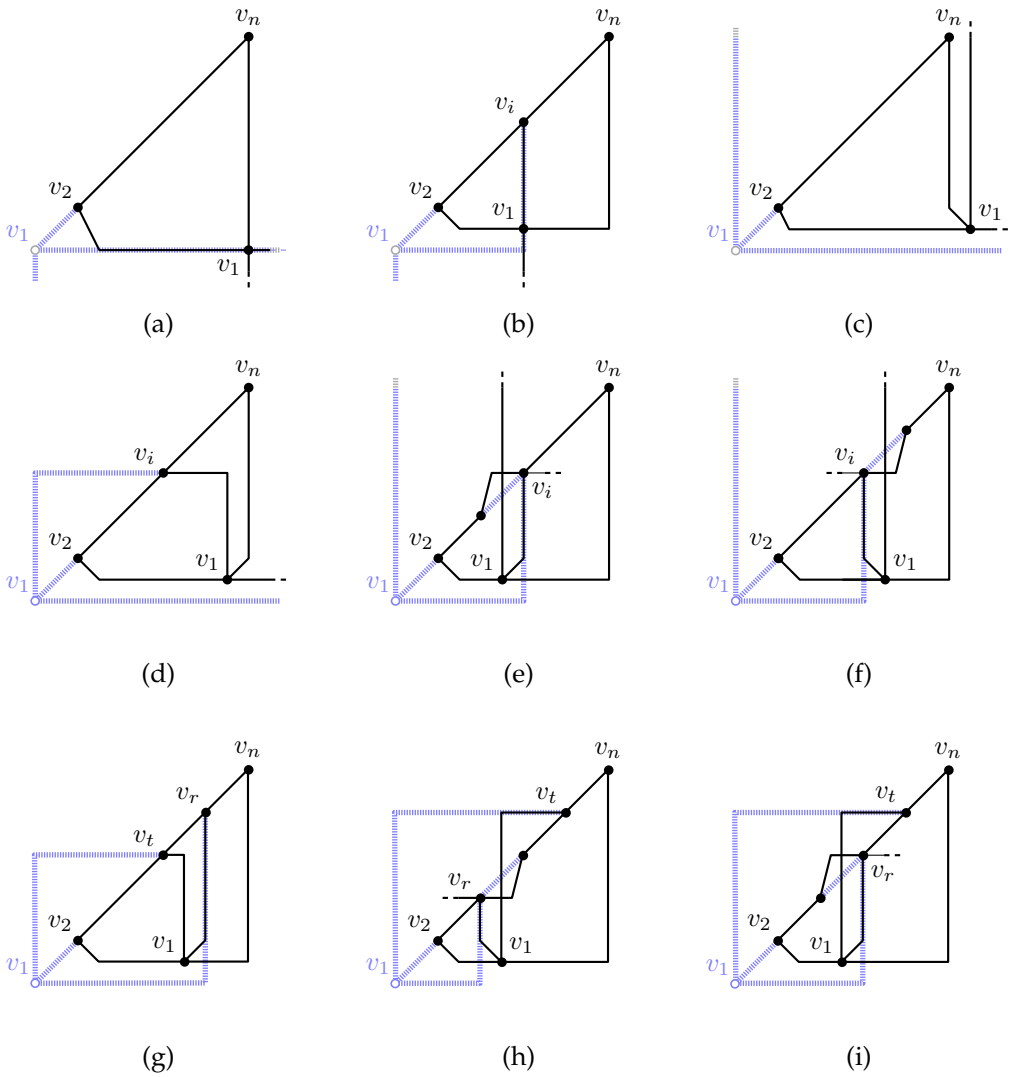


FIGURE 6.4: Illustration of the cases in the proof of Theorem 12 to add the closing edges in C to $\Gamma_{\overline{C}}$. The light blue lines show the initial drawing of edges which are redrawn in order to add the closing edge.

that $s(C) = r(C)$, otherwise we have $s(C) \geq \frac{r(C)}{4}$. Hence it follows that

$$s(F_1) \geq \frac{r(F_1)}{4} \geq \frac{m}{8}$$

edges can be drawn without a bend. □

6.4 2-bend RAC drawings of degree-8 graphs

In the following, we prove that degree-8 graphs admit 2-bend apRAC drawings of quadratic area which can be computed in linear time.

Theorem 13. *Given a degree-8 graph G with n vertices, it is possible to compute in $\mathcal{O}(n)$ time a 2-bend apRAC drawing of G with $\mathcal{O}(n^2)$ area.*

Proof. Let G be a simple degree-8 graph with n vertices. We apply Theorem 9 to augment G to a **directed** 8-regular multigraph having four edge-disjoint 2-factors F_1, F_2, F_3 and F_4, see Fig. 6.5a for an illustrations where G corresponds to K_9. Before we present our algorithm in full detail, we provide a rough outline of the necessary steps.

6.4.1 Outline of the algorithm

In the first step, we will construct two total orders $\prec_x$ and $\prec_y$ of the vertices of G which will determine the x- and y-coordinates of the vertices in the final drawing. In particular, if vertex u of G has the i-th position in $\prec_x$ and the j-th position in $\prec_y$, then u will be placed at point $(8i, 8j)$ in the final drawing. We will construct these two orders independently such that $\prec_x$ is defined by $F_1 \cup F_3$ and $\prec_y$ is defined by $F_2 \cup F_4$. After the computation of $\prec_x$ and $\prec_y$, which finalizes the position of the vertices in our resulting drawing Γ, it remains to draw the edges in order to complete our description of Γ. Every edge will be drawn as a polyline that consists of exactly three segments, which are either horizontal, vertical or oblique. To ensure that all crossings in Γ occur between horizontal and vertical segments, we will restrict oblique segments to be short and require that they are incident to a vertex. To this end, we will define, for each vertex u of G, a closed box $B(u)$ centered at u of size 8×8. Note that by construction, the interior of two boxes do not overlap (they may touch at a corner). Since the x-coordinate of two consecutive vertices u and v of $\prec_x$ differs by exactly 8, there is a vertical line that is (partially) contained inside both $B(u)$ and $B(v)$ (analogous for a horizontal line and consecutive vertices in $\prec_y$). This allows us to join u and v by an edge that consists of two oblique segments, which is called an *oblique*-2 edge. If the unique orthogonal segment of an oblique-2 edge is vertical (horizontal), we will refer to it as a vertical (horizontal) oblique-2 edge. As oblique-2 edges do not occupy an orthogonal-port, we construct $\prec_x$ and $\prec_y$ in order to maximize their number, i.e., consecutive vertices in $\prec_x$ ($\prec_y$) should be connected in $F_1 \cup F_3$ ($F_2 \cup F_4$) as much as possible. An edge that contains exactly one oblique segment will naturally be called an *oblique*-1 edge.

In the second step, we will classify every edge of G as either an oblique-1 or an oblique-2 edge - again this classification is done independently for $F_1 \cup F_3$ and $F_2 \cup F_4$, thus we focus on the description of $F_1 \cup F_3$, the other one is symmetric. Let $e = (u, v)$ be an edge of $F_1 \cup F_3$. If u and v are consecutive in $\prec_x$, then e is classified as a vertical oblique-2 edge. Otherwise, e is classified as an oblique-1 edge such that the (unique) oblique segment is incident to the target v, while the orthogonal segment

at u uses the E-port at u if $u \prec_x v$, otherwise it uses the W-port. Our construction of $\prec_x$ will guarantee that a conflict-free assignment of the ports exists.

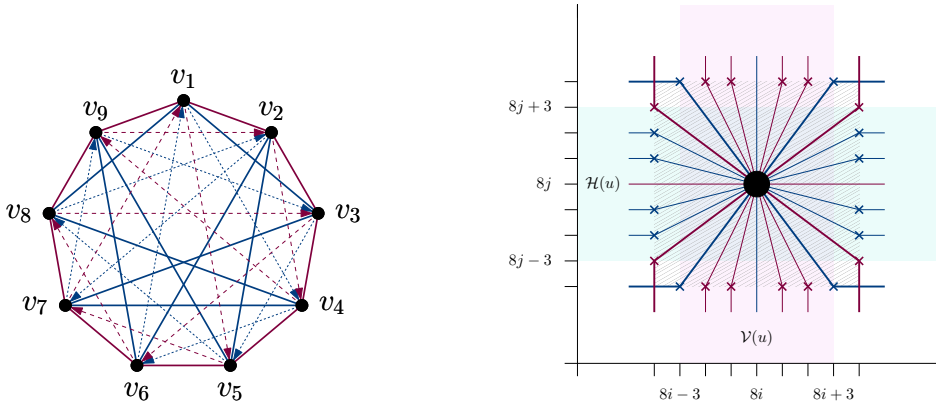
In the final step, we will specify the exact coordinates of the bend-points which is sufficient for a description of the edges. At a high level, oblique segments (which are by construction all incident to vertices) will end at the boundary of the corresponding box, see Fig. 6.5b for a visualization. Since any vertex is incident to at most 8 edges, the size of the box is large enough to accommodate the maximum number of bend-points on each side. The bend-points between vertical and horizontal segments are then naturally defined by the intersections of their corresponding lines.

The final drawing Γ will then satisfy the following two properties.

(i) No bend-point of an edge lies on another edge and

(ii) the edges are drawn with two bends each so that only the edge segments that are incident to u are contained in the interior of $B(u)$, while all the other edge segments are either vertical or horizontal.

This will guarantee that the resulting drawing is 2-bend RAC; for an example see Fig. 6.6. Note that (i) guarantees that no two segments have a non-degenerate overlap.



(a) (b)

FIGURE 6.5: The running example K_9 (a) can be decomposed into four directed edge-disjoint 2-factors; F_1 and F_3 (dashed) are shown in red, while F_2 and F_4 (dotted) are shown in blue. (b) Is an example of the box.

6.4.2 Computing $\prec_x$ and $\prec_y$

We will now describe how to construct $\prec_x$ and $\prec_y$ explicitly. We focus on the construction of $\prec_x$ which is based on F_1 and F_3, the order $\prec_y$ can be constructed analogously. Let $C_1, C_2, \ldots, C_k$ be an arbitrary ordering of the components of F_1. Recall that by definition, each such C_i is a directed cycle. Let S be a set of vertices that contains exactly one arbitrary vertex from each cycle in F_1 and let $P_1, P_2, \ldots, P_k$ be the resulting directed paths obtained by restricting the cycles to $V \setminus S$. Note that this may yield paths that are empty, i.e., when the corresponding cycle consists of a single vertex. We construct $\prec_x$ (limited to $V \setminus S$) such that the vertices of each path appear consecutively defined by the unique directed walk from one endpoint to the other. The relative order between paths is $P_1 \prec_x P_2 \prec_x \cdots \prec_x P_k$. Hence it remains to insert the vertices of S into $\prec_x$. Throughout the algorithm, we will maintain the following invariant which will ensure the correctness of our approach.

I.1 Let $u \in S$ be a vertex of cycle C_i. If $|C_i| > 1$, then u is placed next to at least one of its neighbors in $F_1 \cup F_3$ that belongs to P_i. Otherwise, u is placed directly after the last vertex of C_{i-1} (or as first vertex if $i = 1$) in $\prec_x$.

If Invariant 1 is maintained, we can guarantee the following observation.

Observation 3. *Let $u \in C_i$ and $v \in C_j$ be two vertices of G with $i \neq j$. Then, the relative order of u and v in $\prec_x$ is known.*

Assume that each vertex in S that belongs to $C_1, \ldots, C_{i-1}$ has been inserted in $\prec_x$. Let $u \in S$ be the vertex that belongs to $C_i \setminus P_i$. If $|C_i| \leq 2$, then we place u immediately after the last vertex of C_{i-1} in $\prec_x$ if $i > 1$, otherwise u is the first vertex of $\prec_x$. Hence, for the remainder we can assume that C_i consists of at least three vertices. Let a, b and c be the vertices of G such that (u, a), $(b, u) \in F_1$ and $(u, c) \in F_3$. Even though G is a multigraph, we have that $a \neq b$ since C_i contains at least three vertices. Hence, by construction we have $a \prec_x b$ - in particular, a is the first vertex of P_i in $\prec_x$, while b is the last one. Let C_j (possibly $j = i$) be the cycle that contains c. Note that it is possible that $c \in S$, i.e., c is not part of $\prec_x$ initially. However, as this can only happen if $i \neq j$, we know the relative position of u and c by Observation 3. We distinguish between the following cases based on cycles C_i (which contains u) and cycle C_j (which contains c).

1. $j < i$. We insert u immediately before a in $\prec_x$ such that it is the first vertex of C_i. Clearly, this maintains Invariant 1.

2. $i < j$. We insert u immediately after b in $\prec_x$ such that it is the last vertex of C_i. Clearly, this maintains Invariant 1.

3. $i = j$. In this case, we have that c also belongs to C_i (in particular, c belongs to P_i and thus is already part of $\prec_x$). If $c = a$ or $c = b$, we simply omit the edge (u, c) and proceed as in the first case, i.e., we place u as the first vertex of C_i. Otherwise, we insert u directly before or directly after c in $\prec_x$ based on the edge $(c, d) \in F_3$. The relative order of c and d in $\prec_x$ is known by Observation 3 unless $d \in C_i$. If $d \in P_i$, the relative order between c and d is also known (as both are already present in $\prec_x$). If $d \notin P_i$, then $d = u$ and we can omit the edge $(u, c) \in F_3$ (because it is a copy of $(c, d) \in F_3$), in which case we can again proceed as in the first case. Hence, $d \neq u$ holds. If $c \prec_x d$, we insert u directly before c in $\prec_x$, otherwise we insert u directly after c in $\prec_x$. In both cases, we maintain Invariant 1.

6.4.3 Classification of the edges and port assignment

We focus on the classification of the edges of $F_1 \cup F_3$ and their port assignment, the classification of the edges of $F_2 \cup F_4$ is analogous. Our classification will maintain the following invariants.

I.2 The endpoints of each vertical oblique-2 edge are consecutive in $\prec_x$.

I.3 Each oblique-1 edge $(u, v) \in F_1 \cup F_3$ is assigned the W-port at its source vertex u, if $v \prec_x u$; otherwise, it is assigned the E-port.

I.4 Every horizontal port is assigned at most once.

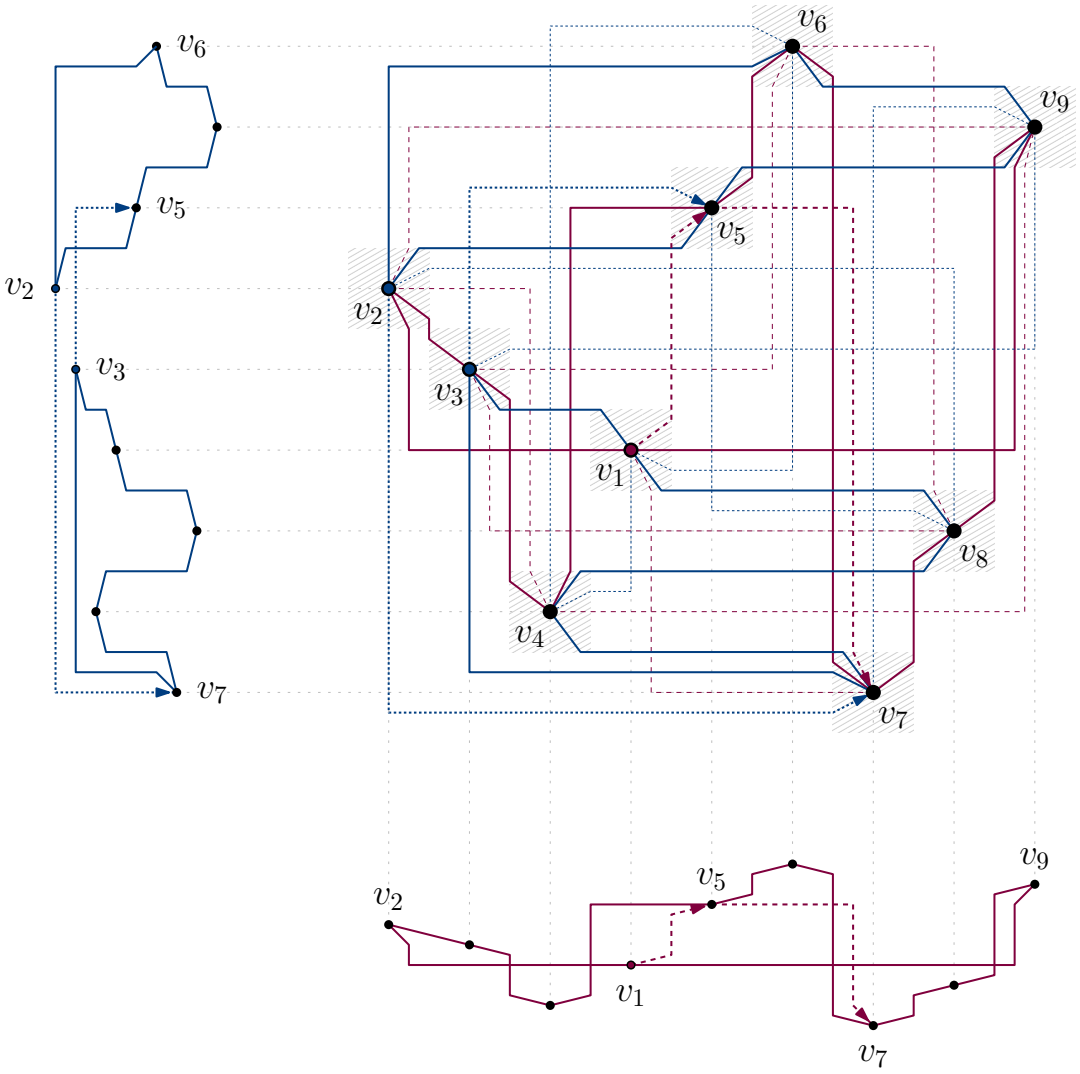


FIGURE 6.6: A 2-bend apRAC drawing of K_9; F_1 and F_3 are red; F_2 and F_4 are blue. Below the drawing of K_9 there is a illustration of the cycles in F_1 and the relevant edges in F_3 for positioning $v_1 \in S$ according to Case 3 in our previous distinction. Similarly, a visualization of the cycles in F_2 and the relevant edges in F_4 is displayed to the left, according to the Cases 1 and 2.

To this end, let us consider an edge e between vertices u and v. If u and v are consecutive in $\prec_x$, then we classify e as a vertical oblique-2 edge. Since all the remaining edges will be classified as oblique-1 edges, this clearly satisfies Invariant 2. If u and v are not consecutive in $\prec_x$, we will classify e as an oblique-1 edge. For any such edge, we will, in an initial phase, assign the ports precisely as stated in Invariant 3. In a subsequent step, we will then create an unique assignment by reorienting some edges of $F_1 \cup F_3$ in order to also guarantee Invariant 4. Hence, suppose that after the initial assignment, there exists a vertex u such that one of its orthogonal ports is assigned twice. Assume first the E-port of u is assigned to edges (u, a) and (u, b). By construction, u has exactly one outgoing edge in F_1, say (u, a), and exactly one outgoing edge in F_3, say (u, b). Let C_i be the cycle of F_1 that contains both u and a (which implies that $|C_i| > 1$) and let C_j be the cycle that contains b (possibly $i = j$). Recall that by construction, the vertices of P_i appear consecutively in $\prec_x$ before the insertion of the vertex $v \in C_i \setminus P_i$. Since u and a are not consecutive in the final $\prec_x$,

we either have $u = v$, $a = v$ or v was inserted directly in between u and a. In the following, we will refer to Case 1 - 3 of Section 6.4.2, where we computed the total order $\prec_x$.

1. $u = v$. Assume first that $C_i \neq C_j$. Then, since $u \prec_x b$ we have $i < j$ and thus according to Case 2, u is placed as the last vertex of C_i in $\prec_x$. This would imply $a \prec_x u$, which is a contradiction to (u, a) using the E-port at u. Hence assume that $C_i = C_j$, i.e., $b \in C_i$. Then we are in Case 3. In particular, we placed u such that u and b are consecutive, thus (u, b) is classified as an oblique-2 edge, again a contradiction.

2. $a = v$. Since $u \prec_x a$, a is not the first vertex of C_i in $\prec_x$, thus we are in Case 2 or in Case 3. If we are in Case 2, then a is placed as the last vertex of C_i. In particular, a is placed next to vertex v' (i.e., the last vertex of C_i) with $(v', a) \in F_1$ by construction. But then the edge that joins u and a in F_1 is directed from a to u, as any vertex has at most one incoming edge in F_1 and we obtain a contradiction.

   If we are in Case 3, then a was placed consecutive to vertex a' with $(a, a') \in F_3$. Moreover, by construction, a has a neighbor in F_1 (i.e., the last vertex of P_i in $\prec_x$) that is after a in $\prec_x$. Hence, we can reorient the edge (u, a) and assign the edge (a, u) the W-port of a conflict-free, which guarantees Invariant 4.

3. v was inserted directly in between u and a. By construction, this only occurs in Case 3. In this instance, v was inserted in between u and a and thus either $(v, u) \in F_3$ or $(v, a) \in F_3$. Suppose first the former. By our construction rule, v is placed after u if there exists a vertex u' such that $(u, u') \in F_3$ and $u' \prec_x u$. But this is impossible, as u' and b cannot coincide, since we have $u' \prec_x u \prec_x b$. Hence, v is placed before a and there exists a vertex a' such that $(a, a') \in F_3$ with $a \prec_x a'$. Further, by construction, a is consecutive to one of its neighbors of F_1 (the one different from u). Hence, we can again reorient e such that $e = (a, u)$ uses the W-port at a to guarantee Invariant 4.

Assume now that the W-port of u is assigned to edges (u, a) and (u, b) with $(u, a) \in F_1$ and $(u, b) \in F_3$. The following case analysis might look symmetric to the previous one, however there exist some subtle differences. Again, let C_i be the cycle of F_1 that contains both u and a (which implies that $|C_i| > 1$) and let C_j be the cycle that contains b (possibly $i = j$). Since u and a are not consecutive in the final $\prec_x$, we either have $u = v$, $a = v$ or v was inserted directly in between a and u which gives rise to the following cases.

1. $u = v$. Assume first that $C_i \neq C_j$. Then, since $b \prec_x u$ we have $j < i$ and thus according to Case 1, u is placed as the first vertex of C_i in $\prec_x$. But then $u \prec_x a$ holds and hence (u, a) is assigned the W-port at u.

   Hence assume that $C_i = C_j$, i.e., $b \in C_i$. Then we are in Case 3. In particular, we placed u such that u and b are consecutive, thus (u, b) is classified as an oblique-2 edge, again a contradiction.

2. $a = v$. Since $a \prec_x u$, a is not the last vertex of C_i in $\prec_x$, thus we are in Case 1 or in Case 3. If we are in Case 1, then a is placed as the first vertex of C_i since there exists a vertex a' with $a' \prec_x a$ such that $(a, a') \in F_3$. Moreover, a is placed next to vertex v' (i.e., the first vertex of C_i) with $(a, v') \in F_1$ by construction. This allows us to redirect the edge $(u, a) \in F_1$ such that we can assign (a, u)

the E-port at a which solves the conflict at u and thus guarantees Invariant 4. If we are in Case 3, then a was placed consecutive to vertex a' with $(a, a') \in F_3$. Moreover, by construction, a has a neighbor in F_1 (i.e., the first vertex of P_i in $\prec_x$) that is before a in $\prec_x$. Hence, we can reorient the edge (u, a) and assign the edge (a, u) the E-port of a conflict-free, which guarantees Invariant 4.

3. v was inserted directly in between a and u. By construction, this only occurs in Case 3. In this instance, v was inserted in between a and u such that $a \prec_x v \prec_x u$ holds and thus either $(v, u) \in F_3$ or $(v, a) \in F_3$. Suppose first the former holds, i.e., $(v, u) \in F_3$. By our construction rule, v is placed before u if there exists a vertex u' such that $(u, u') \in F_3$ and $u \prec_x u'$. But this is impossible, as u' and b cannot coincide, since we have $b \prec_x u \prec_x u'$. Hence, v is placed after a and there exists a vertex a' such that $(a, a') \in F_3$ with $a' \prec_x a$. Further, by construction, a is consecutive to one of its neighbors of F_1 (the one different from u). Hence, we can again reorient e such that $e = (a, u)$ uses the E-port at a to guarantee Invariant 4.

Observe that if an edge (u, v) was redirected, then both u and v belong to the same cycle C_i of F_1 and since this operation has to be performed at most once per cycle, it follows that they can be considered independently. So far, we have computed $\prec_x$ and classified every edge of $F_1 \cup F_3$ guaranteeing Invariants 1-4. Symmetrically, we can compute $\prec_y$ and classify every edge of $F_2 \cup F_4$ guaranteeing corresponding versions of Invariants 1-4.

I.5 Let $\mathcal{S}$ be a set containing exactly one arbitrary vertex from each of the cycles $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_\kappa$ of F_2 and denote by $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_\kappa$ the resulting paths when restricting the cycles to $V \setminus \mathcal{S}$. Let $u \in \mathcal{S}$ be a vertex of cycle $\mathcal{C}_i$. If $|\mathcal{C}_i| > 1$, then u is placed next to at least one vertex of $\mathcal{P}_i$ in $\prec_y$. Otherwise, u is placed directly after the last vertex of $\mathcal{C}_{i-1}$ (or as first vertex if $i = 1$) in $\prec_y$.

I.6 The endpoints of each vertical oblique-2 edge are consecutive in $\prec_y$.

I.7 Each oblique-1 edge $(u, v) \in F_2 \cup F_4$ is assigned the S-port at its source vertex u, if $v \prec_y u$; otherwise, it is assigned the N-port.

I.8 Every vertical port is assigned at most once.

6.4.4 Bend placement

We begin by describing how to place the bends of the edges on each side of the box $B(u)$ of an arbitrary vertex u based on the type of the edge that is incident to u, refer to Fig. 6.5b. Let (x_u, y_u) be the coordinates of u in Γ that are defined by $\prec_x$ and $\prec_y$. Recall that the box $B(u)$ has size 8×8. Let e be an edge incident to u. We focus on the case in which $e \in F_1 \cup F_3$, the other case in which e belongs to $F_2 \cup F_4$ is handled symmetrically by simply exchanging x with y, "top/bottom" with "right/left" and "vertical" with "horizontal" from the following description. By definition, e is either an oblique-1 edge or a vertical oblique-2 edge. Suppose first that e is an oblique-1 edge. If $e = (u, v)$, i.e., e is an outgoing edge of u in $F_1 \cup F_3$, then by Invariant 3 edge e uses either the W- or E-port at u. In the former case, the segment of e incident to u passes through point $(y_u, y_u - 4)$, while in the latter case it passes through point $(y_u, y_u + 4)$. For an example, refer to the outgoing edge (v_3, v_6) of v_3 in Fig. 6.6. If $e = (v, u)$, i.e., e is an incoming edge of u in $F_1 \cup F_3$, then by Invariant 3 e uses a horizontal port at v and by the fact that every edge consists of

exactly three segments, the vertical segment of e ends at the top or the bottom side of $B(u)$. Since any vertex has at most three incoming edges in $F_1 \cup F_3$ by construction, we can place the respective bends at x-coordinate $x_u + i$ with $i \in \{-2, -1, 1, 2\}$ and y-coordinate $y_u + 4$ ($y_u - 4$) for the top (bottom) side such that the assigned i-value is unique, refer to the incoming edge (v_4, v_9) of v_9 in Fig. 6.6, where $i = -1$. Once the i-value is determined, the other bend-point of e is uniquely defined as it connects a vertical with a horizontal segment by construction. Namely, this bend-point is at $(x_u + i, y_v)$.

Suppose now that e is a vertical oblique-2 edge. By Invariant 2, u and v are consecutive in $\prec_x$. If $v \prec_x u$ the x-coordinate of the bend point is $x_u - 4$, otherwise it is $x_u + 4$; e.g., refer to the edges (v_2, v_3) and (v_3, v_4) of v_3 in Fig. 6.6, respectively. In order to define the y-coordinate of the bend point, we have to consider the relative position of u and v in $\prec_y$. If $v \prec_y u$ the y-coordinate of the bend point of e is $y_u - 3$ and otherwise it is $y_u + 3$. Invariant 2 implies that any vertex has at most two vertical oblique-2 edges since no vertex has more than two direct neighbors in $\prec_x$. From the description of the bend-points, we can deduce the following observation.

Observation 4. *Let b be a bend-point that delimits an oblique segment s which belongs to an edge e. If s is incident to u, then b does not lie on any other edge incident to u.*

6.4.5 Proof of correctness

We will show that Γ satisfies Properties (i) and (ii), which then immediately guarantee that Γ is a 2-bend apRAC drawing. To this end, we will introduce some additional notation. For an edge $e = (u, v)$, we will denote by b_u the first bend-point encountered when traversing e starting at its source u, while the second bend-point is denoted by b_v. Fix a vertex u and assume it is the i-th vertex in $\prec_x$ and the j-th vertex in $\prec_y$. By construction, u has coordinates $(8i, 8j)$ in Γ. We define by $\mathcal{H}(u)$ the horizontal strip $[8i - 3, 8i + 3]$ and by $\mathcal{V}(u)$ the vertical strip $[8j - 3, 8j + 3]$ of u; see Fig. 6.5b for an illustration. By our construction we obtain the following results.

Proposition 1. *Any vertical segment that belongs to an oblique-1 edge (u, v) is completely contained in $\mathcal{V}(u)$ or $\mathcal{V}(v)$. Similarly, any horizontal segment that belongs to an oblique-1 edge (u, v) is completely contained in $\mathcal{H}(u)$ or $\mathcal{H}(v)$.*

Proof. W.l.o.g. assume that $e = (u, v) \in F_1 \cup F_3$, the other case is symmetric. By construction, the horizontal segment is delimited by u and b_u, whose coordinates are either $(x_u - 4, y_u)$ or $(x_u + 4, y_u)$, thus the horizontal segment is completely contained inside $\mathcal{H}(u)$. Moreover, b_v has coordinates $(x_v + i, y_v \pm 4)$ with $i \in \{-2, -1, 1, 2\}$ and thus the vertical segment delimited by b_u and b_v is completely contained inside $\mathcal{V}(v)$ as desired. □

Proposition 2. *For every vertex $u \in G$, any vertical (horizontal) segment contained in $\mathcal{V}(u)$ ($\mathcal{H}(u)$) belongs to an edge which is incident to u.*

Proof. First observe that by construction, no box $B(v)$ with $v \neq u$ is contained inside $\mathcal{V}(u)$ or $\mathcal{H}(u)$. We will provide the proof of the statement for $\mathcal{V}(u)$, the one for $\mathcal{H}(u)$ is symmetric. Suppose for a contradiction that $\mathcal{V}(u)$ contains a vertical segment s which belongs to an edge $e = (v, w)$ such that $v \neq u \neq w$. Suppose first that e is an oblique-2 edge. Clearly, since a horizontal oblique-2 edge does not contain a vertical segment, we can assume that e is a vertical oblique-2 edge. By Invariant 4, this implies that v and w are consecutive in $\prec_x$. Segment s of e overlaps with the boundary of both $B(v)$ and $B(w)$ by construction - but then it cannot be contained inside $V(u)$,

as $V(u)$ contains neither $B(v)$ nor $B(w)$ by the initial observation. Suppose now that $e = (v, w)$ is an oblique-1 edge. If $e \in F_2 \cup F_4$, then s is incident to v by construction and intersects the top/bottom side of $B(v)$. Otherwise, if $e \in F_1 \cup F_3$, b_2 lies on the bottom/top side of $B(w)$. Since $v \neq u \neq w$ holds and no other box besides $B(u)$ is contained inside $\mathcal{V}(u)$, we obtain a contradiction. □

Proof that Property (i) is maintained in Γ. We leverage these propositions to show that our drawing maintains Property (i). Again, we fix an edge $e = uv$ and distinguish based on the type e.

1. $e = (u, v)$ is an oblique-1 edge and b_u or b_v lie on another edge.

   W.l.o.g. assume that $e \in F_1 \cup F_3$ and that $u \prec_x v$. Then, edge e uses the E-port at u by Invariant 3 and b_v is placed on the top/bottom side of $B(v)$ (depending on the relative order of u and v in $\prec_y$) by construction. Since b_v is contained inside $\mathcal{V}(v)$ and since b_u and b_v are connected by a vertical segment, it follows that also $b_u \in \mathcal{V}(v)$. Now, any vertical segment contained in $\mathcal{V}(v)$ belongs to an edge that is incident to v by Proposition 2 - but then Observation 4 guarantees that no overlap between such a segment and b_v can occur and hence neither with b_u. Symmetrically, a horizontal segment overlapping b_u is inside $\mathcal{H}(u)$ and hence belongs to an edge that is incident to u by Proposition 2. By Observation 4, we can then ensure no overlap between a horizontal segment and b_u. Finally, consider a horizontal segment which overlaps b_v. If this segment would belong to an oblique-1 edge (u', v'), then it is contained in $\mathcal{H}(u')$ if $(u', v') \in F_1 \cup F_3$ or in $\mathcal{H}(v')$ otherwise. But as b_v lies on the boundary of $B(v)$, we would have that $B(v)$ is contained in either $\mathcal{H}(u')$ or $\mathcal{H}(v')$, a contradiction. Hence, this segment needs to belong to an oblique-2 edge $\{u, v\}$; in particular to a horizontal one. By construction and Invariant 4, u' and v' are consecutive in $\prec_y$ and the segment is contained in both $B(u')$ and $B(v')$. If $v = u'$ or $v = v'$, then by Observation 4 no overlap occurs. Thus, since $u' \neq v \neq v'$, we have that $b_v \in H(v)$ and neither $B(u')$ nor $B(v')$ are contained inside $H(v)$, so no overlap can occur.

2. $e = uv$ is an oblique-2 edge and b_u or b_v lies on another edge.

   Recall that by construction, b_u and b_V lie on the boundary of $B(u)$ and $B(v)$, respectively. Observation 4 establishes that they do not lie on edge-segments, in particular oblique ones, whose corresponding edges are incident to u or v. Since no two boxes overlap, it follows that b_u and b_v do not lie on any other oblique segment. Suppose now that b_u or b_v lie on a vertical or horizontal segment which belongs to an oblique-1 edge (u', v'). But as $b_u \in B(u)$ and $b_v \in B(v)$, and since these segments are always contained in the vertical or horizontal strip of either u' or v', which does not contain the box of another vertex, such an overlap would only be possible when u, v, u' and v' are not pairwise disjoint. But then again, Observation 4 asserts that no such overlap can exist. Finally, we assume that b_u or b_v lie on the middle part of another oblique-2 edge $e' = u'v'$. If both e and e' are vertical/horizontal, then there exists a pair of vertices with the same x-coordinate/y-coordinate, which is impossible by construction. Hence, w.l.o.g. assume that e is a vertical oblique-2 edge, while e' is a horizontal one such that b_u lies on e', the other case is symmetric. By construction, the y-coordinate of b_u is $y_u \pm -3$ - hence the horizontal

segment of e' would be contained in $\mathcal{H}(u)$ which is by Proposition 2 only possible if $u = u'$ or $u = v'$ holds, but then we can use Observation 4 to guarantee that there is no overlap.

Proof that Property (ii) is maintained in Γ. It remains to show that Property (ii) is satisfied. Consider the first part of Property (ii),i.e., only the edge-segments that are incident to a vertex u are contained in the interior of $B(u)$. Suppose for a contradiction that there exists a segment s that is partially contained inside $B(u)$ such that s is not incident to u. Since by construction, any oblique segment is fully contained inside a box and since no two boxes overlap, s cannot be an oblique segment. Suppose first that s belongs to an oblique-2 edge $e = vw$ in which case s is incident to neither v nor w. Segment s lies on the boundary of both $B(v)$ and $B(w)$ and cannot pass through the interior of another box, as otherwise v and w would not be consecutive in $\prec_x$ or $\prec_y$, a contradiction to Invariant 4 or Invariant 6. Suppose now that s belongs to an oblique-1 edge (v, w). Then, s is contained inside the vertical/horizontal strip of either v or w by Proposition 1, which does not contain $B(u)$ and we obtain a contradiction. This establishes the first part of Property (ii). For the latter part, we have to show that our description of bend points guarantees that the segments outside of the boxes are in fact vertical or horizontal. To be more precise, we have to show for any segment s with endpoints a and b (which can be bend points or vertices), that a and b differ only in x- or y-coordinate. Consider an edge $e = uv$. Assume first that e is a vertical oblique-2 edge such that $u \prec_x v$. Then u and v are consecutive in $\prec_x$ by Invariant 2. By construction, we placed b_u at x-position $x_u + 4$, while the x-position of b_v is $x_v - 4$. As x and y are consecutive in $\prec_x$, we have $x_v = x_u + 8$ and hence $x_u + 4 = x_v - 4$, thus the segment is indeed vertical. Analogously, we can argue for the case where e is a horizontal oblique-2 edge. Finally, suppose that $e = (v, w)$ is an oblique-1 edge. Again, we describe the case where $e \in F_1 \cup F_3$ in detail, the other is analogous. W.l.o.g. assume that $v \prec_x w$. By Invariant 3 and construction, the segment of e incident to v uses the E-port of e. Consider a horizontal ray that is incident to e to the right. Clearly, we can find a point on this ray, such that a perpendicular ray to the first one either intersects either top or the bottom side $B(w)$ at its assigned i-value (depending on the relative order of v and w in $\prec_y$). This intersection point then uniquely defines b_i and Property (ii) is ensured.

Space and time complexity To complete the proof of Theorem 13, we discuss the time complexity and the required area. We apply Theorem 9 to G to obtain F_1, F_2, F_3 and F_4 in $\mathcal{O}(n)$ time, as the input graph has bounded degree. For each cycle of F_1 and F_2, an appropriate ordering of its internal vertices, the classification of the incident edges and the assignment of the orthogonal ports can be computed in time linear in the size of the cycle. Clearly, computing the bend-points can be done in linear time in the number of edges as well. Hence we can conclude that the drawing can be computed in $\mathcal{O}(n)$ time. For the area, we can observe that the size of the grid defined by the boxes is $8n \times 8n$ and by construction, any vertex and any bend point is placed on a distinct point on the grid, which therefore yields quadratic area.

□

Remark We remark here that the produced drawing is not necessarily simple, i.e, it is possible that two edges have more than one point in common. In particular, it is possible that two adjacent edges cross or that two edges cross even twice. For

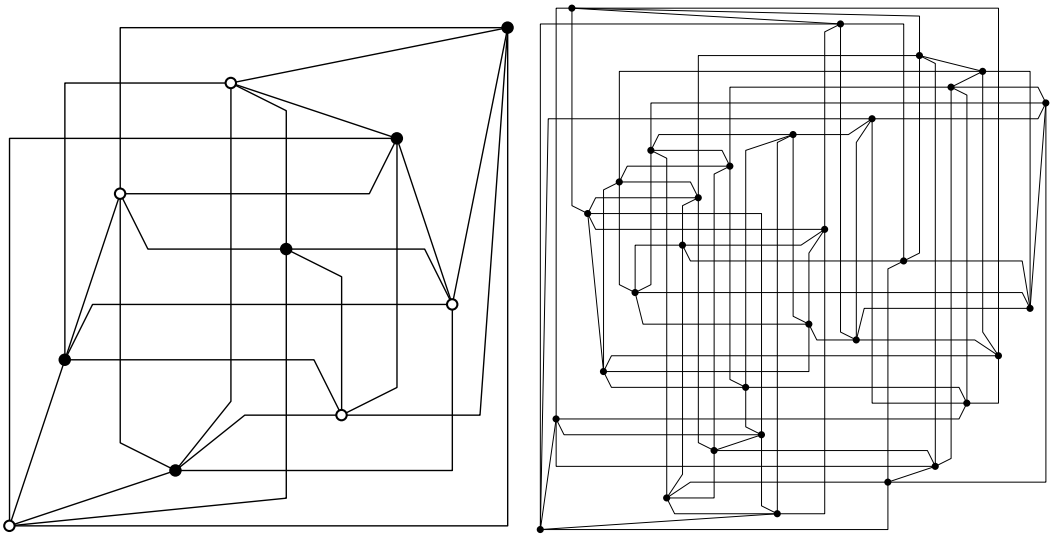
most of the vertices, this could be solved in a postprocessing step (e.g., by assigning the i-values in Subsection 6.4.4 carefully). However, for vertices of S (refer to Section 6.4.2)) it is not immediate if such a postprocessing is possible with our approach and we will consider it as an open problem whether every degree-8 graph admits a simple 2-bend RAC drawing. To this end, we remark that [7] showed that the requirement of a simple drawing is a real restriction on the class of graphs in the 1-bend RAC setting, i.e., they provided an infinite family of graphs which admit non-simple 1-bend RAC drawings but do not admit simple 1-bend RAC drawings. In fact, even the maximum edge-density of these two variants differs.

6.5 Open problems

Most of the open problems that arise by our work are the extensions of our results to higher degrees. To this end, let us consider some upper bounds for the values which we could attain. As discussed in the introduction of this chapter, there exists degree four graphs which are not RAC. For 1- and 2-bend RAC the situation is not as simple. Since there are no negative results for small graphs, we can only work with the known edge-density results. In particular, 1-bend RAC graphs have at most $5.5n - 11$ edges[7] and since K_{10} has $45 > 5.5 \cdot 10 - 11$ edges, it follows that degree-8 is best possible. A similar bound could be obtained for 2-bend RAC graphs, but since the upper bound is presumably far from tight, the result is not really interesting. Hence, we post the following open problems.

1. Are all degree-3 graphs RAC? By the previous discussion, this result would be best possible.

2. Are all degree-5 graphs 1-bend RAC? Fig. 6.7 provides 1-bend RAC drawings of two prominent degree-5 graphs. The left side is a drawing of $K_{5,5}$, the right side is the 5-cube graph. Note that already for these rather regular graphs, we could not identify a pattern which can be used to generalize to arbitrary degree-5 graphs.



(a) (b)

FIGURE 6.7: 1-bend RAC drawings for (a) the $K_{5,5}$ graph, and (b) the 5-cube graph.

3. Are all degree-9 graphs 2-bend RAC? In the following, we will highlight that such an extension is highly non-trivial. Let G be a degree-9 graph. Assume for now we do not have to consider closing edges which can be achieved if G contains two edge-disjoint Hamiltonian Cycles. Then we would have that every vertex has exactly one vertical and exactly one horizontal port free. Further, we assume that G is 9-edge colorable, i.e., we can partition the edges into four 2-factors and a perfect matching M. But even then, for an edge $ab \in M$, we can not guarantee that one of the free ports at a or b can be used for the edge of M, since this depends on their relative order in $\prec_x$ and $\prec_y$ which is fixed by the traversal of the Hamiltonian Cycle.

Part II

Structural Research

Chapter 7

Axis-parallel RAC drawings

In the previous chapter, we investigated the containment of bounded degree graphs in the class of k-bend RAC graphs for small values of k. In fact, in order to control the crossing-angles in our constructions, we only allowed crossings between segments that are either parallel or perpendicular to the x-axis. We remark here that this behavior is not unique to our algorithms, as also the constructions in [13, 15, 72] admit such restricted versions of RAC drawings. Motivated by this observation, we introduce the class of apRAC drawings in this chapter and consider structural properties such as the relationship to general RAC drawings, edge-density results or the complexity of the recognition problem.

Related work The work of [74] initiated the study of RAC drawings and showed that RAC graphs with n vertices have at most $4n - 10$ edges, while bipartite RAC graphs have at most [10] $3n - 7$ edges. The recognition of RAC graphs was shown to be NP-hard [16]; in fact, it is $\exists\mathbb{R}$-complete even when the embedding is fixed and even if every edge is crossed at most ten times [161]. Besides the straight-line setting, RAC drawings were also studied when bends are allowed. In particular, [74] showed that every graph admits a 3-bend RAC drawing. Regarding the density, 1-bend RAC graphs can have at most $5.5n - 11$ edges [7], and, interestingly, if we require the graphs to admit simple drawings, then the corresponding bound is only $5.4n - \mathcal{O}(1)$ [7]. Finally, n-vertex 2-bend RAC graphs can have at most $20n - 24$ many edges [174].

Our contribution We initiate the theoretical study of apRAC drawings which form a natural subset of RAC drawings both in the absence of bends and when bends are allowed. In Section 7.2 we are concerned with 0-bend apRAC graphs - we establish that an n-vertex 0-bend apRAC graph has at most $4n - \sqrt{2}\sqrt{n} - 6$ edges and provide an almost matching lower bound construction with $4n - 2\sqrt{n} - 7$ edges. Since there exists n-vertex RAC graphs with $4n - 10$ edges [74], it follows that there is a proper containment relation between 0-bend apRAC graphs and RAC graphs. Motivated by this, we identify the smallest graph which is RAC but not 0-bend apRAC. Finally, we translate the recognition result from general RAC graphs, i.e., we show that it is still NP-hard to decide if a graph is 0-bend apRAC.

In Section 7.3 we prove an upper bound of $5n - 6$ edges on the number of edges of an n-vertex 1-bend apRAC graph and provide a lower-bound construction with $5n - 13$ edges. Again, since there exists n-vertex 1-bend RAC graphs with $5.5n - \mathcal{O}(1))$ edges [7], we have a proper containment relationship.

In Section 7.4 we show that n-vertex 2-bend apRAC graphs have at most $10n - 12$ edges, which is tight up to an additive constant as we provide a construction with $10n - 46$ edges. We highlight here that this construction, which is also 2-bend RAC by definition, is a substantial improvement over the current best lower-bound

for general 2-bend RAC graphs that contains $7.62n - \mathcal{O}(1)$ many edges [17]. We further derive improved upper bounds on the edge density when restricting the corresponding graph class to be bipartite. Finally, Section 7.5 introduces a natural extension of the apRAC setting.
This chapter is based on joint work with Patrizio Angelini, Michael A. Bekos, Julia Katheder, Michael Kaufmann and Torsten Ueckerdt which is accepted at the"European Symposium on Algorithms (ESA) 2023"; a preliminary version was accepted at the "European Workshop on Computational Geometry" [9].

7.1 Preliminaries

We begin by stating some elementary properties of 0-bend (ap)RAC drawings. Properties 7 and 8 hold for 0-bend RAC (and thus for 0-bend apRAC) drawings.

Property 7 ([74]). *In a 0-bend RAC drawing no edge is crossed by two adjacent edges.*

Property 8 ([74]). *A 0-bend RAC drawing does not contain a triangle T formed by edges of the graph and two edges ab and ab', such that a lies outside T and b, b' lie inside T.*

The following two properties are limited to 0-bend apRAC drawings.

Property 9. *A 0-bend apRAC drawing does not contain a triangle T formed by edges of the graph and three vertices v_1, v_2, v_3 incident to a vertex u, such that v_1, v_2, v_3 lie outside T and u lies inside T.*

Proof. Assume the contrary. Then, by Property 7, no two edges incident to u can cross the same boundary edge of T - but then every boundary edge of T is crossed and thus T consists of three axis-parallel segments, which can not be completed to a triangle, hence we obtain a contradiction. □

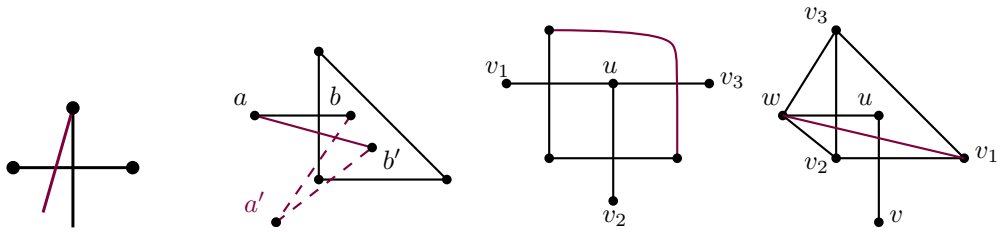
Property 10. *Let Γ be a 0-bend apRAC drawing containing a triangle T formed by edges of the graph and two adjacent vertices u and v such that u is contained inside T while v is outside T. Then, Γ cannot contain a vertex incident to u, v and all vertices of T.*

Proof. Suppose for a contradiction that vertex w is incident to u, v and all vertices of T. If w is contained inside T, then we have exactly the setting of Property 8 with $a = v$, $u = b$ and $w = b'$, a contradiction. Thus, w is outside T. Since vu and vw both cross the boundary of T, we have by Property 7 that T is a right-angle triangle with two axis-parallel legs. W.l.o.g. assume that v_1v_2 and v_2v_3 are the legs of T such that v_1v_2 is horizontal, while v_2v_3 is vertical. W.l.o.g. we further assume that v_2v_3 is crossed by the edge uw. Then, since wv_1 is present, it follows that v_2v_3 is crossed by uw and wv_1, which is impossible by Property 7, and we obtain a contradiction. □

In Theorems 16 and 20 we leverage the following property shown in [16] of the so-called *augmented square antiprism graph*, depicted in Fig. 7.4a.

Property 11 ([16], Theorem 1). *Any straight-line RAC drawing of the augmented square antiprism graph has two combinatorial embeddings.*

Property 12 ([16], Theorem 2). *The horizontal or vertical extension of two augmented square antiprism graphs has a unique combinatorial embedding in the straight-line RAC setting.*



(a) (b) (c) (d)

FIGURE 7.1: Illustrations of Property 7 - Property 10. Red edges are impossible.

7.2 0-bend apRAC graphs

In this section, we present an almost tight bound on the edge-density of 0-bend apRAC graphs. Further, we establish the minimal example graph which separates 0-bend apRAC from 0-bend RAC graphs.

Theorem 14. *A 0-bend apRAC graph with n vertices has at most $4n - \sqrt{n} - 6$ edges. In fact, there are not infinitely many values of n for which there exists n-vertex 0-bend apRAC graphs with more than $4n - \sqrt{2}\sqrt{n} - 6$ edges.*

Concerning the lower bound, there exist infinitely many 0-bend apRAC graphs with n vertices and $4n - 2\lfloor\sqrt{n}\rfloor - 7$ edges.

Proof. We will first show that n-vertex 0-bend apRAC graphs have at most $4n - \sqrt{n} - 6$ edges for any n. Consider an n-vertex 0-bend apRAC graph G and its corresponding 0-bend apRAC drawing Γ. We can w.l.o.g assume that Γ uses at least $\sqrt{n}$ different x-coordinates (if Γ would use less than $\sqrt{n}$ different coordinates in each direction, we could not fit the n vertices on the implicit grid). Consider the subgraph G_v of G defined by the set E_v of vertical edges of Γ. G_v is a forest of paths and contains at least one component for every x-coordinate of Γ - hence at least $\sqrt{n}$ components in total. As the number of edges of a an n-vertex forest with c components is $n - c$, it follows that $|E_v| \leq n - \sqrt{n}$. Every edge of $E \setminus E_v$ is either drawn horizontally or oblique in Γ, thus $G - E_v$ is crossing-free and has at most $3n - 6$ edges, whence G has at most $4n - \sqrt{n} - 6$ edges as desired.

Now, for the second part, assume for a contradiction that there exists infinitely many n for which there exists n-vertex 0-bend apRAC graph with at least $4n - \alpha\sqrt{n} - 6$ for some $\alpha < \sqrt{2}$. Let $G = (V, E)$ be such a graph with $|V| = n$ and let Γ be a 0-bend apRAC drawing of G. We will partition the edges of G into the disjoint sets E_v, E_h and E_o. The edges of E_v correspond to vertical segments in Γ that are involved in a crossing, the edges of E_h correspond to horizontal segments in Γ that are involved in a crossing and the remaining edges are the one of E_o. By construction, we observe that the subgraphs $G - E_v = E_o \cup E_h$ as well as $G - E_h = E_o \cup E_v$ are planar and hence contain at most $3n - 6$ edges. Thus, $|E| \leq 3n - 6 + \max\{|E_v|, |E_h|\}$ and hence by assumption on G we have $\max\{|E_v|, |E_h|\} \geq n - \alpha\sqrt{n}$. Denote by C the number of crossings in Γ. By construction, every crossing involves a horizontal and a vertical segment, thus $C \geq \max\{|E_v|, |E_h|\} \geq n - \alpha\sqrt{n}$. On the other hand, denote by L_v and L_h the set of vertical and horizontal lines that contain the vertices of G in Γ. Now consider the induced subgraph $G_v = (n, E_v)$. Every component of G_v forms a path that lies on the same vertical line of L_v (note that it is possible that we have several components on the same vertical line). Thus, G_v is a forest of paths with at least $|L_v|$ components and hence $|E_v| \leq n - |L_v|$. Using a similar argument,

we establish that $|E_h| \leq n - |L_h|$. Finally, as every vertex and crossing lie on the intersection of a vertical line of L_v and a horizontal line of L_h by construction, we have $|L_h| \cdot |L_v| \geq n + C$. Hence we obtain

$$\begin{aligned} & 2n - \alpha\sqrt{n} \leq n + C \leq |L_v| \cdot |L_h| \leq (n - |E_v|) \cdot (n - |E_h|) \leq \alpha\sqrt{n} \cdot \alpha\sqrt{n} \\ \Leftrightarrow \quad & (2 - \alpha^2)n \leq \alpha\sqrt{n} \end{aligned} \tag{7.1}$$

Now for $\alpha < \sqrt{2}$ the lefthand side of (7.1) is $\Omega(n)$ while the righthand side is $O(\sqrt{n})$; clearly, for sufficiently large n this yields a contradiction.

For the lower bound, consider the construction shown in Fig. 7.2. For any even $k > 0$, construct a $k \times k$ grid graph H_k which contains a pair of crossing edges in every quadrangular face. Let G_k be the graph obtained from H_k by adding two extremal adjacent vertices W and E connected to $2k - 1$ consecutive boundary vertices of H_k each (refer to the blue edges in Fig. 7.2 and observe that the edge between W and E can be added by moving W and E upwards of H_k). If we denote by n the number of vertices of G_k, then $n = k^2 + 2$, $k = \sqrt{n-2}$ and thus $m = 4n - 2\lfloor\sqrt{n}\rfloor - 7$.

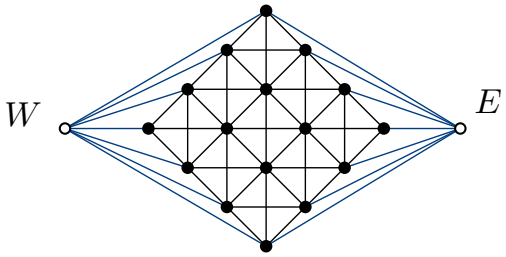


FIGURE 7.2: Lower bound construction for 0-bend apRAC

□

Corollary 5. *A bipartite 0-bend apRAC graph with n vertices has at most $3n - \sqrt{n} - 4$ edges.*

Proof. The proof is analogous with the only crucial observation that, since a subgraph of a bipartite graph is necessarily bipartite, we have that $|E| \leq 2n - 4 + \max\{|E_v|, |E_h|\}$ which yields the desired result. □

Corollary 6. *The class of 0-bend apRAC graphs is a proper subclass of the one of 0-bend RAC graphs. Likewise, the class of bipartite 0-bend apRAC graphs is a proper subclass of the one of bipartite 0-bend RAC graphs.*

Proof. There exists (bipartite) 0-bend RAC graphs on n vertices with $(3n - 7)$ $4n - 10$ edges. Hence, the corollary directly follows from the density bounds of Corollary 5 and Theorem 14. □

In Theorem 15, we show that K_6 minus one edge is the minimal separating example; i.e., K_6 minus one edge is 0-bend RAC but not 0-bend apRAC.

Theorem 15. *Graph $K_6 - e$ is the minimal example separating the classes of 0-bend RAC and 0-bend apRAC.*

Proof. Let $G = K_6 - e$. Fig. 7.3a, which is a 0-bend RAC drawing, establishes that G is indeed RAC. Suppose for a contradiction that G admits a 0-bend apRAC drawing Γ.

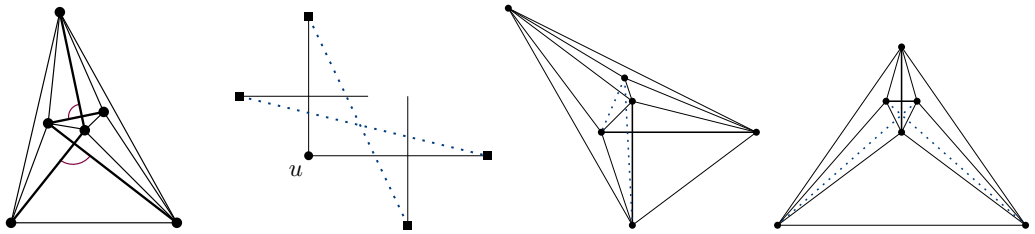


FIGURE 7.3: Illustrations for the proof of Theorem 15.

Since G has 6 vertices and 14 edges and since $cr(G) \geq |E[G]| - (3|V[G]| - 6)$ holds (as a planar graph G can have at most $3|V[G]| - 6$ edges), it follows that $cr(G) \geq 14 - 12 = 2$. Since the drawing consists of straight-line edges, any crossing involves exactly four different vertices. Hence, by the pigeonhole principle, there exists a vertex u of G which is incident to (at least) two crossing edges uv and uw in Γ. Denote by e_1 and e_2 the edges that cross uv and uw in Γ, respectively. Assume first that v, u and w are colinear. By Property 7, the endpoints of e_1 and e_2 are necessarily distinct and different from u,v and w. Since G only contains six vertices, we obtain a contradiction. Hence, one of uv and vw is horizontal, while the other is vertical and we are in the setting of Fig. 7.3b. Since G is a complete graph minus one edge, at least one of the blue edges is present which is impossible by Property 7. Hence, we conclude that G is not 0-bend apRAC.

To show the minimality of G, we first observe that K_5 admits a RAC drawing with exactly one crossing [74], which is therefore also 0-bend apRAC for an appropriate orientation of the axis. Finally, we will provide 0-bend apRAC drawings for the two non-isomorphic graphs obtained by removing one additional edge from $G = K_6 - e$. In Fig. 7.3c two adjacent edges were removed from K_6, while in Fig. 7.3d two independent edges were removed from K_6. □

We conclude this section by studying the recognition problem of whether a graph is 0-bend apRAC.

Theorem 16. *It is NP-hard to decide whether a given graph is 0-bend apRAC.*

Proof. In order to prove the statement, we adjust the NP-hardness reduction (from 3-SAT) for the general case of straight-line RAC graphs introduced in [16]. Based on the so-called *augmented square antiprism graph*, which by Property 11 has two combinatorial embeddings in the RAC setting, the construction of the clause-gadgets, the variable-gadgets as well as the connections between them is based on a basic *building block* having the following properties: (i) It has a unique embedding, (ii) there are four vertices properly contained in its interior, which can be connected to vertices in its exterior by crossing a single boundary edge, (iii) no edge can (completely) pass through it without forming a fan crossing and (iv) it can be extended horizontally or vertically maintaining the aforementioned properties. Unfortunately, even though the augmented square antiprism graph is in fact 0-bend apRAC, the building block of [16] is not. In the following, we prove that the graph G of Fig. 7.4c satisfies properties $(i) - (iii)$; thus, it can act as the building block for our reduction. Observe that G is composed of an augmented square antiprism graph H and a 4-cycle C connected to the blue vertices of H. Recall that, by Property 11, H has two combinatorial embeddings $\mathcal{E}_1$ and $\mathcal{E}_2$ in the 0-bend RAC setting and thus at most two in the 0-bend apRAC setting; refer to shown in Figs. 7.4a and 7.4b, respectively. Since only axis-parallel edges are involved in crossings and since each crossing-free edge of $\mathcal{E}_1$ and

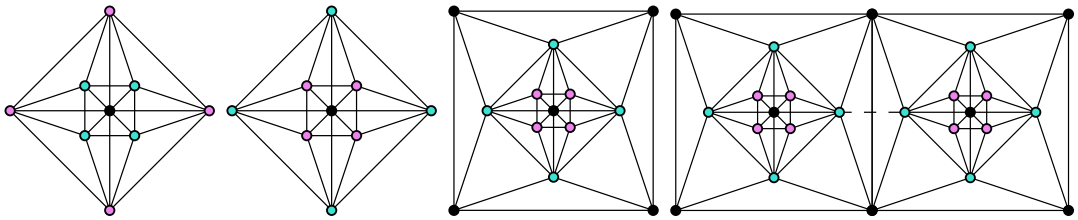


FIGURE 7.4: Illustrations for the proof of Theorem 16.

$\mathcal{E}_2$ is not axis-parallel, it follows that the crossing-free edges of $\mathcal{E}_1$ and $\mathcal{E}_2$ cannot be crossed in a 0-bend apRAC drawing of G. This implies that the four blue vertices of H which are connected to C have to lie on a common face of the subgraph of H induced by the crossing-free edges in $\mathcal{E}_1$ or $\mathcal{E}_2$. This is impossible in $\mathcal{E}_1$ and unique in $\mathcal{E}_2$, as illustrated in Fig. 7.4c. This is enough to guarantee (i). Further, the figure clearly asserts that (ii) and (iii) are also guaranteed. Property (iv) can be guaranteed in the exact same way as in the original paper [16], see Fig. 7.4d. Since our building block is 0-bend apRAC and since any crossing that does not involve a building block appears between axis-parallel edges in the original reduction, it follows that the constructed drawing is 0-bend apRAC if and only if the input 3-SAT formula is satisfiable. □

7.3 1-bend apRAC graphs

In this section, we will establish an upper bound and an almost matching lower bound for the class of 1-bend apRAC graphs.

Theorem 17. *A 1-bend apRAC graph with n vertices has at most $5n - 8$ edges. Also, there exist infinitely many 1-bend apRAC graphs with n vertices and $5n - 13$ edges.*

Proof. For the upper bound, consider a 1-bend apRAC drawing Γ of an n-vertex graph G. Each edge segment in Γ is either horizontal (h), vertical (v) or oblique (o). For $x, y \in \{h, v, o\}$, let E_{xy} be the edges of G with two edge segments of type x and y. Then, E_{hv}, E_{ho}, E_{vo} and E_{oo} form a partition of the edge-set of G, assuming that edges that consist of only one h-, v- or o-segment are counted towards E_{ho}, E_{vo} and E_{oo}, respectively. By construction, any crossing involves exactly one vertical and one horizontal segment. Hence, the subgraph of G induced by $E_{ho} \cup E_{oo}$ is planar and contains at most $3n - 6$ edges. Further, as every segment is incident to a vertex and since any vertex is incident to at most two vertical segments, we have $|E_{vo} \cup E_{hv}| \leq 2n$. Consider now the vertical edge-segment incident to the topmost vertex v_t that uses the N-port (if it is present). By construction, it cannot be involved in a crossing, as otherwise it would cross a horizontal edge-segment whose endpoints would lie above v_t, a contradiction to the choice of v_t. Thus, this segment can be replaced by a steep oblique edge-segment without introducing new crossings. Analogous observations can be made for the bottommost vertex in Γ, which implies that $|E_{vo} \cup E_{hv}| \leq 2n - 2$. Thus, $|E| = |E_{ho}| + |E_{vo}| + |E_{hv}| + |E_{oo}| \leq 5n - 8$.

For the upcoming lower bound construction, refer to Fig. 7.6 for an illustration of the base construction with $n = 20$. For arbitrary n, we arrange $n - 4$ vertices in the shape of a triangle with an empty base such that there are an (almost) equal number of vertices on each side which are joined by a y-monotone path (of size $n - 7$) in addition to the cycle of size $n - 4$ (inner black edges in Fig. 7.6). All $n - 4$

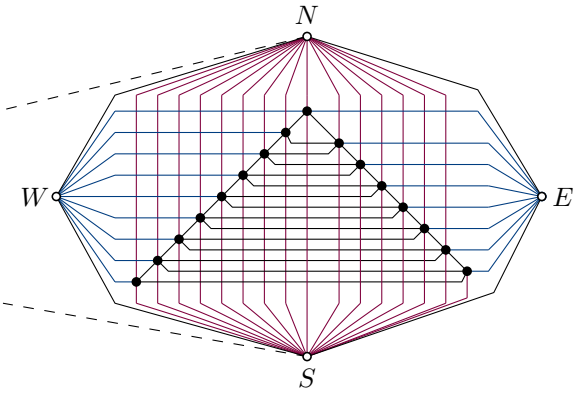


FIGURE 7.5: Basic lower bound construction for the class of 1-bend apRAC graphs.

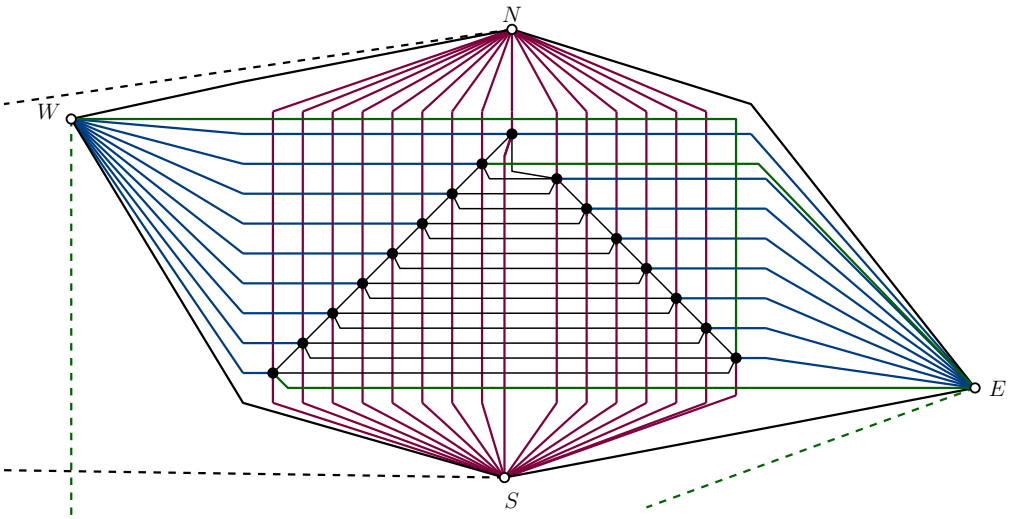


FIGURE 7.6: Augmented construction for the lower bound. Note that the dotted rays of the same color will intersect and thus define 1-bend edges

(interior) vertices are connected to two universal vertices N and S above and below the triangle (edges in purple), while the universal vertices W and E to the left and to the right of the triangle are connected only to the inner vertices which lie on their respective side of the triangle (edges in turquoise). The exception is the vertex placed at the top of the triangle, which can be connected to both W and E. Finally, one can add five edges between the universal vertices (edges in black) which gives a total of $n-4+n-7+3(n-4)+1+5 = 5n-17$ edges for the base construction. With a more elaborate positioning of the extremal vertices, one can add four additional edges to obtain the desired bound of $5n-13$ edges, refer to Fig. 7.6.

□

Corollary 7. *Bipartite* 1*-bend apRAC graphs with n vertices have at most* $4n-6$ *edges.*

Since there exists 1-bend RAC graphs with $5.5n-72$ edges [7], the following corollary is immediate.

Corollary 8. *The class of* 1*-bend apRAC graphs is a proper subclass of the one of* 1*-bend RAC graphs.*

7.4 2-bend apRAC graphs

In Theorems 18 and 19 we give upper and lower bounds on the edge-density of 2-bend apRAC graphs, respectively.

Theorem 18. *A 2-bend apRAC graph with n vertices G has at most $10n - 12$ edges.*

Proof. Consider a 2-bend apRAC drawing Γ of an n-vertex graph G. Each edge-segment in Γ is either horizontal (h) or vertical (v) or oblique (o). Denote by S the set of edges that contain at least one segment in {h,v} incident to a vertex. Since any vertex is incident to at most two vertical and at most two horizontal segments, it follows that $|S| \leq 4n$. Let E_h, E_v and E_o be the set of edges of $E \setminus S$ whose middle-part is h, v and o, respectively. Assuming that an edge of $E \setminus S$ consisting of less than three segments belongs to E_o, it follows that E_h, E_v and E_o form a partition of $E \setminus S$. Observe that the edges of E_o cannot be involved in any crossing in Γ, as all of its segments are oblique. Further, no two edges of E_h or E_v can cross. Hence, the subgraphs induced by $E_h \cup E_o$ and $E_v \cup E_o$ are planar and contain at most $3n - 6$ edges each. Recall that $|S| \leq 4n$ and thus $|E| \leq |S| + |E_h| + |E_v| + 2|E_o| \leq 4n + 3n - 6 + 3n - 6 = 10n - 12$. □

Corollary 9. *Bipartite 2-bend apRAC graphs with n vertices have at most $8n - 8$ edges.*

Theorem 19. *There exist infinitely many 2-bend apRAC graphs with n vertices and $10n - 46$ edges.*

Proof. For the following construction, we refer to Fig. 7.7 for an illustration - a full example can be found in Fig. 7.8.

Fix an integer $k \geq 6$ and consider a set S of k^2 points that form a $k \times k$ square grid. However, the grid is not axis-parallel, but slightly rotated, say counterclockwise, such that the points in each column have consecutive x-coordinates, while the points in each row have consecutive y-coordinates. We will define two distance-functions. Namely, $dist_x(p, q)$ is the number of points whose x-coordinates lie in between the x-coordinates of p and q, while $dist_y(p, q)$ is the number of points whose y-coordinates lie in between the y-coordinates of p and q. Then we obtain the following important property by construction.

$$\text{For any } p \neq q \in S \text{ we have } \mathrm{dist}_x(p, q) + \mathrm{dist}_y(p, q) \geq k - 1 \geq 5. \tag{7.2}$$

To see that this holds, suppose for a contradiction that there exists a pair of points $p, q \in S$ such that $\mathrm{dist}_x(p, q) + \mathrm{dist}_y(p, q) < k - 1$ holds. W.l.o.g. assume that p belongs to the i-th row and the j-th column of our grid. If q does not belong to row $i - 1, i$ or $i + 1$, then by construction we already have $\mathrm{dist}_y(p, q) >= k - 1$. At the same time, q has to belong to column $j - 1, j$ or $j + 1$. Since $p \neq q$, q differs from p in at least one of row and column. If it differs in exactly one, then the distance is k by construction - if it differs in both, we have distance $k + 1$, hence we obtain a contradiction.

Between any $p, q \in S$ which are consecutive in x-coordinate, i.e., $dist_x(p, q) = 0$, we will add a 2-bend edge whose middle-segment is vertical and that has two (short) oblique segments incident to its endpoint. Analogously, we add a 2-bend edge whose middle-segment is horizontal between any pair of vertices that is consecutive in y-coordinate. Clearly, we do not introduce any multiple edges by this construction as no two vertices are consecutive in x- and y-coordinate by (7.2). In total, we have $2(k^2 - 1)$ such edges.

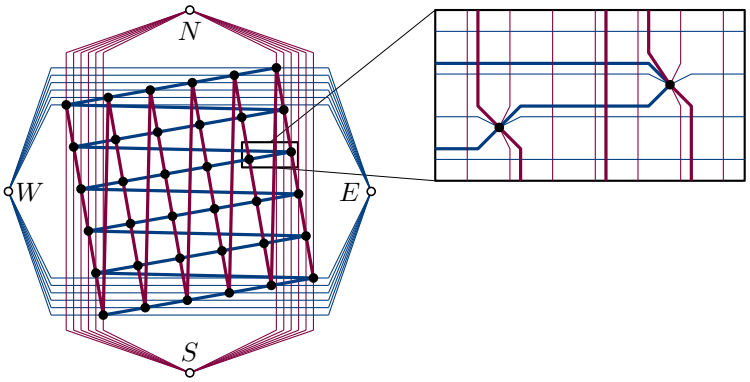


FIGURE 7.7: Illustration of the construction in Theorem 19 with $k = 6$. Edges with two slanted segments are indicated in pink for vertical middle segment and in blue for horizontal middle segments. Edges using the horizontal or vertical ports are omitted for readability.

For the next step, we will add four universal vertices N, E, S and W to the top, right, bottom and left of all points of S, respectively. Every point of p will be connected to N by a 2-bend edge that starts with a short oblique segment at p, followed by a vertical middle-segment which ends with an oblique segment (which is almost horizontal) at N, consider the red edges in Fig. 7.8 for an illustration. Similarly, we add an edge from every point p to S using a vertical middle-segment and to each of W and E an edge with a horizontal middle-segment (blue in Fig. 7.8). In total, we added $4k^2$ many edges such that only the middle-segments of the edges are involved in a crossing (to see the local routing of the short oblique segments, consider the right side of Fig. 7.7). Hence, so far we did not use any orthogonal port at the points of S. We will leverage these free ports to add four more 2-bend edges for (almost) each point of S. First, consider for each point $p \in S$ the point $q \in S$ below p such that $dist_y(p, q) = 1$. We draw a 2-bend edge from p to q that starts with a vertical segment at p which uses the S-port of p. This segment almost reaches the y-coordinate of q, where we continue with a horizontal middle-segment until we almost reach the x-coordinate of q and we end the construction of the edge with a short oblique segment incident to q (this segment can be made arbitrarily short as we are close in x- and y-direction after the previous two segments). Similarly, we use the N-port of p for an edge to point q' that is above p with $dist_y(p, q') = 2$. Here, we use distance two in order to avoid the introduction of a parallel edge. Symmetrically, we will draw the two edges at p which use the horizontal ports. Recall that we add these edges to almost all points of S. In particular, if point p belongs to the two bottommost vertices of S, then the point q below p such that $dist_y(p, q) = 1$ is undefined. Moreover, if p belongs to the three topmost vertices, then the point q' above p such that $dist_y(p, q') = 2$ holds is undefined. Hence, we loose five edges per direction, and we added a total of $4k^2 - 10$ edges. Again, we can observe that the only crossings that occur are due to horizontal and vertical segments. To complete the description, it is easy to create a K_4 on the universal vertices without interfering with the remainder of the drawing. In summation, we have constructed a 2-bend apRAC drawing with $n = k^2 + 4$ vertices and

$$2(k^2 - 1) + 4k^2 + (4k^2 - 10) + 6 = 10k^2 - 6 = 10n - 46$$

edges, which concludes the proof.

□

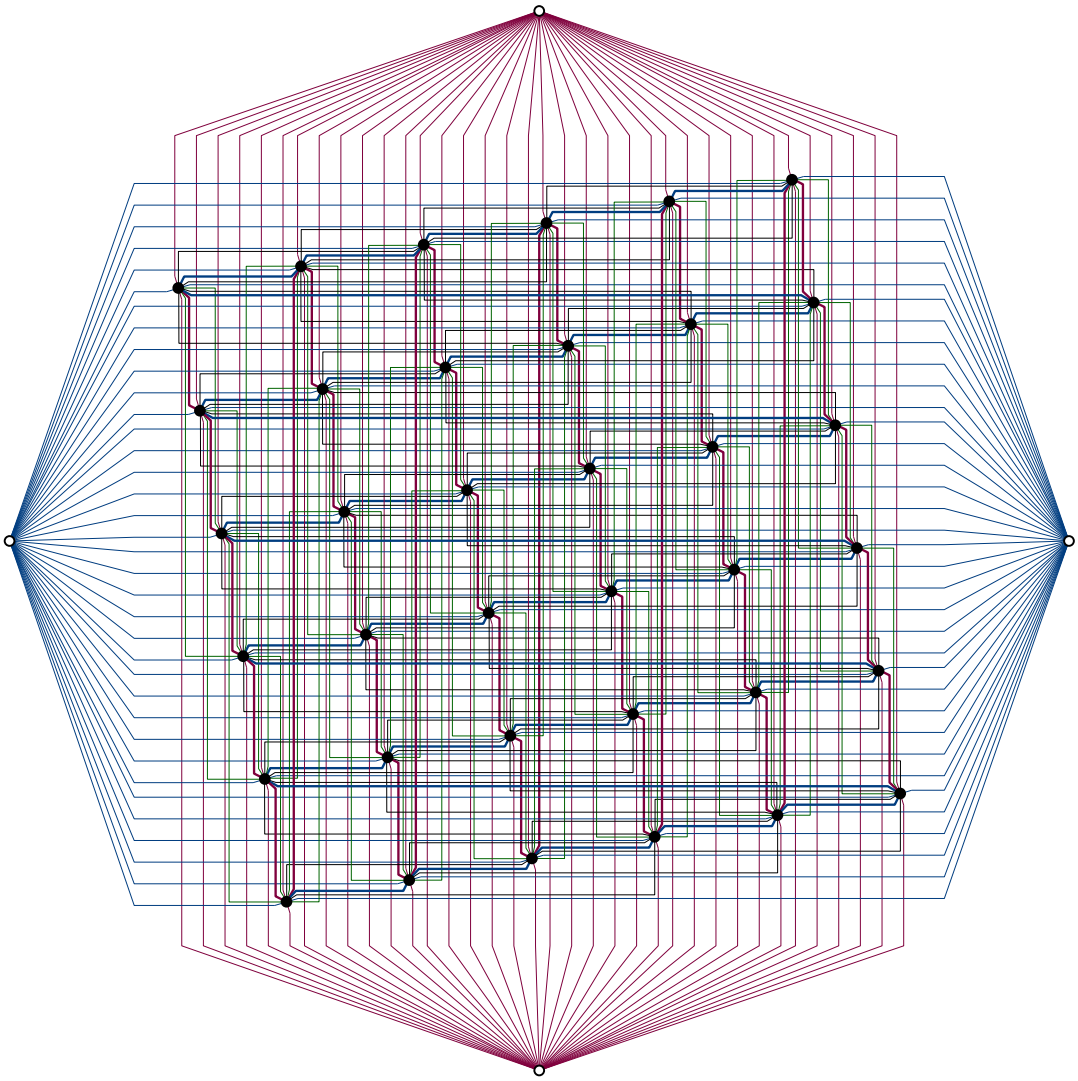
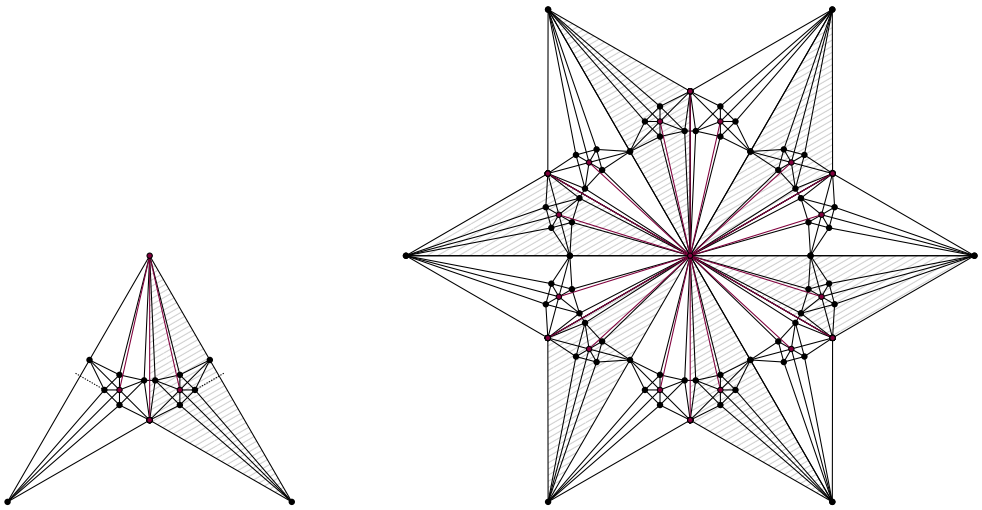


FIGURE 7.8: Full example for the construction in Theorem 19 .

7.5 Generalization of apRAC

A natural generalization of apRAC drawings is a drawing where each edge segment involved in a crossing is parallel or perpendicular to a line having one out of s different slopes. We denote the class of graphs admitting such a drawing with k bends as *k-bend s-RAC graphs*. Clearly, k-bend 1-RAC graphs coincide with the previously covered k-bend apRAC graphs. In the following, we establish a proper inclusion of 0-bend s-RAC graphs with respect to the general 0-bend RAC graphs for any sublinear s.



(a) (b)

FIGURE 7.9: (a) Graph G which has a unique combinatorial embedding in the 0-bend RAC setting by Property 12 (b) Graph G_6 used in Theorem 20

Theorem 20. *There exist 0-bend RAC graphs on n vertices which are not 0-bend s-RAC for any $s \in o(n)$.*

Proof. Let G be the extension of two augmented square antiprism graphs, which has a unique combinatorial embedding by [16]; see Fig. 7.9a. The final graph G_k consists of k copies of G, namely $G_1, \ldots, G_k$ such that G_i and G_{i+1} (modulo k) are connected by an additional edge (red in Fig. 7.9b) and all copies of G share one vertex V (center in Fig. 7.9b). In Fig. 7.9b a RAC drawing of G_6 is shown. Clearly, this drawing can be extended to any $k \mod 2 = 0$. Since the red edges form a horizontal extension the embedding of G_k is unique. But then vertex v is necessarily incident to $2k$ crossing edges whose angle formed with the x-axis is pairwise different. Hence, in order to admit a 0-bend s-RAC drawing, s has to be at least $\frac{k}{2}$. But clearly, $k \in \Omega(n)$, a contradiction. □

We now turn our attention to 2-bend s-RAC graphs. In particular, we establish an upper bound dependent on s which is better than the one for general 2-bend RAC graphs for small s.

Theorem 21. *A 2-bend s-RAC graph G with n vertices has at most* $\min\{(6+4s)n - 12, 20n - 24\}$ *edges.*

Proof. For $s \geq 4$, the current best upper bound for 2-bend RAC graphs [174] holds. Therefore let G be an optimal 2-bend s-RAC graph and Γ a valid drawing of G with

$s < 4$. There are at most four edge segments incident to every vertex for every line l of unique slope, namely two edge segments parallel to l and two edge segments perpendicular to l. Hence, in total we have at most $4sn$ many edge segments incident to the n vertices that can be involved in a crossing by definition. Denote the edges which contain at least one such segment as I. Clearly, $|I| \leq 4sn$ and $E \setminus I$ is a set of edges where only the middle segment of the edges can be involved in any crossing. Denote by D the subdrawing of Γ restricted to the edges of $E \setminus I$. We claim that the intersection graph $\mathcal{I}$ of D is bipartite. Before we proof the claim, we will show that it yields the desired result. Namely, since $\mathcal{I}$ is bipartite, there exists a two coloring such that in D, no two edges of the same color intersect. Thus, the subdrawing of D restricted to each of the two colors is planar and contains at most $3n - 6$ edges, hence $|E \setminus I| \leq 6n - 12$ and thus $|E| = |E \setminus I| + |I| \leq 6n - 12 + 4sn = (6 + 4s)n - 12$ as desired. It remains to proof the claim. Recall that only the middle segments of the edges of $E \setminus I$ are involved in crossings in D. Subdividing the edges at the bends and then restricting D to only contain the edges which corresponded to middle segments in D yields a straight-line RAC drawing D' such that the crossing graph of D and of D' coincide. Since D' is a 0-bend RAC drawing, we have that the crossing graph of D' is bipartite [76], and hence the crossing graph of D is as well, which concludes the proof. □

7.6 Open problems

We state the following open problem raised by our work.

1. Close the gap of the edge-density (at least up to an additive constant) for the case of 0-bend apRAC graphs. We conjecture that the provided lower bound is best possible (up to an additive constant).

2. The proofs for the upper bound of the edge-density for $k \in \{1, 2\}$ were not concerned with the simplicity of the drawing - does the maximum edge-density differ significantly if we enforce the drawing to be simple, such as in the case of 1-bend RAC drawings [7]? At least for $k = 1$, our lower bound construction (which is a simple drawing) whose number of edges is tight up to a small constant indicates that the bounds may be the same - however, our lower bound construction for $k = 2$ contains many adjacent edges that intersect (and it is not obvious how to resolve this issue). Hence, a natural question would be if there exists graphs which admit a k-bend apRAC drawing, but not a k-bend apRAC simple drawing for $k \in \{1, 2\}$.

3. Do the classes of 2-bend apRAC and 2-bend RAC coincide?

4. What is the edge-density of 2-bend RAC graphs? In particular, is it larger than the one of 2-bend apRAC graphs?

5. Is the recognition problem for k-bend apRAC graphs hard for $k \in \{1, 2\}$? This is most likely true, however most known reduction techniques require gadget graphs with a unique embedding, which is difficult to guarantee in the presence of bends.

Chapter 8

Bipartite gap-planar graphs

Introduction This chapter is concerned with the study of k-gap-planar graphs. The motivation for this beyond-planar graph class is twofold. On the one hand, it can be seen as an asymmetric version of k-planarity, as any crossing is charged to exactly one of the two edges which are involved in the crossing in the k-gap-planar setting, while the crossing is charged to both edges in the k-planar one. On the other hand, k-gap-planarity is motivated by a drawing paradigm which was developed in order to avoid the visual clutter introduced by many edge-crossings in straight-line drawings. The rough idea is to simply omit the part of the edge-segment where the crossings occurs to improve readability. So called partial edge drawings (*PED*) [48] are drawings where the middle-part of every edge-segment is omitted, such that the remaining segments (which are also called stubs) that are incident to the vertices are crossing-free. While this paradigm is optimal w.r.t. the number of crossings, the omitted middle-parts often form the majority of the edge-segments, which unfortunately impairs readability. To deal with this issue while still avoiding crossings, the method of *edge casing* was introduced in [86]. In a cased (straight-line) drawing, every edge-crossing is solved by locally interrupting one of the two edge-segments which forms the crossing, see the left side of Fig. 8.1 for an example. The idea is that such an interruption only impairs the readability of the interrupted edge. Hence, if the problem that we try to visualize yields a ranking of importance on the edges, we can ensure that important edges are readable even if they are involved in many crossings (by simply interrupting all the other edges which cross the important ones).

Related work k-gap-planar graphs were introduced in [21]. The authors studied inclusion relationships with other beyond-planar graph classes (in particular, they showed that every $2k$-planar graph is k-gap-planar, while the converse is not necessarily true). Further, they established a tight density bound for the case of general 1-gap-planar graphs and provided an upper bound on the edge-density of k-gap-planar graphs with the use of the crossing lemma. Bipartite 1-gap-planar graphs were considered by [162], who gave an upper bound of the edge-density which depends on the number of so called *critical components* of the crossing graph, which can be linearly (in the number of vertices) many.

Our contribution In the following, we will drop this dependence on the number of critical components and show that an n-vertex bipartite 1-gap planar graph has at most $4n - 8$ edges. In fact, we will show a slightly stronger statement, namely that n-vertex bipartite 1-gap-planar non-homotopic multigraphs contain at most $4n - 8$ edges. This chapter is part of the joint work with Aaron Büngener titled "On the edge density of bipartite 3-planar and bipartite gap-planar graphs" which was

recently accepted at the "32nd International Symposium on Graph Drawing and Network Visualization (GD2024)".

8.1 Edge-density of bipartite gap-planar graphs

In the remainder of this chapter, we will abbreviate 1-gap-planarity with gap-planarity. Denote by $\mathcal{G}$ the set of all tuples (G, Γ) where G is a bipartite graph of n vertices and Γ is a gap-planar drawing of G where any two copies of an edge are non-homotopic. Let $\mathcal{G}' \subset \mathcal{G}$ consist of all elements (G, Γ) such that G has the maximum number of edges. Finally, let $\mathcal{G}'' \subset \mathcal{G}'$ consist of all elements (G, Γ) such that Γ has the least number of crossings. For the remainder, we fix such a tuple $(G, \Gamma) \in \mathcal{G}''$.

We leverage the following result which holds for any 1-gap planar drawing, hence in particular for Γ.

Lemma 20 ([21]). *The crossing graph $\mathcal{I}$ of Γ is a pseudoforest.*

In order to show that G has at most $4n - 8$ edges, we want to find a set of edges of G, denoted by P, such that no two edges of P are crossing in Γ and such that $|P| \geq \frac{m}{2}$.

The result then follows immediately since P induces a bipartite planar subgraph of G, hence $|E[P]| \leq 2n - 4$ and thus $|E[G]| \leq 4n - 8$ follows. In order to define P, we will consider the individual components of the intersection graph $\mathcal{I}$. Recall that by definition, no two edges that belong to different components of $\mathcal{I}$ do intersect, thus we can consider the components separately.

Lemma 21. *Let X be an arbitrary component of $\mathcal{I}$. X contains an independent set of size at least $\frac{|X|}{2}$ if X*

- *is a tree,*
- *contains an even-length cycle or*
- *contains an odd-length cycle C and at least one path (that is edge-disjoint from C) of odd length rooted at a vertex of C.*

Proof. In the first two cases, any cycle (if it exists) has even length by Lemma 20, hence X is bipartite and it admits a 2-coloring. Clearly, both colors are independent sets by definition, and one of the color classes contains at least $\frac{|X|}{2}$ nodes. For the third case, let $C = (v_1, v_2, \ldots, v_k)$ and w.l.o.g. assume that the path $(u_1, u_2, \ldots, u_{2j+1})$ is rooted at v_1 such that $v_1u_1 \in X$. Coloring u_i with i odd implies that we have colored $j + 1$ vertices in one color, which is enough to accommodate for all vertices of the path in addition to v_1. Clearly, the only vertex that is not on the path which is influenced by the coloring is v_1. Removing v_1 (and the whole path) from X yields a tree since C is the unique cycle of X, which has an independent set of the desired size as shown in the previous case. Combining both independent sets then concludes the proof. □

The remainder of the components that are not included in the aforementioned cases will be called *critical components*. These components are pseudotrees, i.e., trees that contain exactly one cycle - which in our case is necessarily odd and any path rooted at the vertices of the cycle has even length (the size of such a path can be 0). For the critical components, we can not directly find an independent set in $\mathcal{I}$ of appropriate size. To be more precise, we can only find an independent set of size $\frac{|X|}{2} - 1$. To overcome this issue, we will show that for any such component, there exists sufficiently many uncrossed edges in Γ (which correspond to singletons in $\mathcal{I}$).

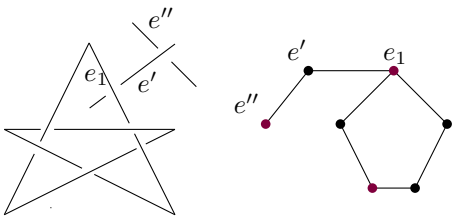


FIGURE 8.1: Example of a critical component. The maximum independent set (vertices in red) is not sufficiently large.

Lemma 22. *Let $e_1 = a_1b_1$ and $e_2 = a_2b_2$ be two edges of G that intersect in Γ. Then, either a_1b_2 or a_2b_1 is present in G drawn along e_1 and e_2.*

Proof. Suppose for a contradiction that neither a_1b_2 nor a_2b_1 exist in G drawn along e_1 and e_2. Denote by x the intersection point between e_1 and e_2. W.l.o.g. assume that the crossing between e_1 and e_2 was assigned to e_1 in Γ. This implies that no crossing on the (open) segments a_1x and xb_1 can be assigned to e_1. Moreover, at most one crossing is assigned to e_2 by definition - w.l.o.g. assume this crossing is due to an edge that intersects e_2 on the segment a_2x. Now, consider the graph $G' = G \setminus \{a_1b_1, a_2b_2\} \cup \{a_1b_2, a_2b_1\}$ with corresponding drawing Γ' where the drawing of all edges but a_1b_2 and a_2b_1 is inherited from Γ, while the edges a_1b_2 and a_2b_1 are drawn along (the original curves of) e_1 and e_2, refer to Fig. 8.2a. First observe that G' is a non-homotopic multigraph, as neither a_1b_2 nor a_2b_1 drawn along e_1 and e_2 were present by assumption. Further, Γ' is a valid gap-planar drawing as we do not need to assign a_1b_2 any crossing, while a_2b_1 is assigned at most one crossing. But then we obtain a contradiction to our choice of G and Γ, as G' contains the same number of edges as G, but Γ' contains strictly less crossings than Γ. □

We can now turn our attention to the critical components. As we will argue about graph G and its intersection graph $\mathcal{I}$ simultaneously, we will assume in the following that an edge e_i of G corresponds to a vertex v_i of $\mathcal{I}$. Let X be a critical component in $\mathcal{I}$ and let $C = (v_1, v_2, \ldots, v_{2j+1})$ be its unique odd cycle in $\mathcal{I}$. Pick two adjacent vertices v_1 and v_2 of C and assume that $e_1 = a_1b_1$ and $e_2 = a_2b_2$. Recall that Lemma 22 guarantees that one of a_1b_2 or a_2b_1 exists in G such that its curve follows e_1 and e_2 in Γ. W.l.o.g. assume that a_1b_2 exists and denote by v the corresponding vertex of a_1b_2 in $\mathcal{I}$. We distinguish between the following two cases based on whether a_1b_2 intersects an edge of C or not.

Assume first that a_1b_2 intersects an edge of C. As a_1b_2 is drawn along e_1 and e_2, this edge is either e_{2j+1} or e_3 by construction. Note that a_1b_2 cannot intersect both of these edges, as otherwise X is not a pseudoforest. W.l.o.g assume that a_1b_2 crosses e_3, the other case is symmetric. If there is an additional edge e' besides e_3 that is crossing a_1b_2, then e' also crosses either e_1 or e_2 as a_1b_2 is drawn along e_1 and e_2, but then X is not a pseudoforest as this crossing would close another cycle. In particular, if it crosses e_1, then we obtain the cycle $(e_1, e_2, e_3, a_1b_2, e')$, and otherwise we obtain (e_2, e_3, a_1b_2, e'). Hence, a_1b_2 only crosses e_3 - but then we have an odd-length path rooted at v_3 in $\mathcal{I}$ (that only contains vertex v), in which case X is not critical, a contradiction.

Thus we can assume from now on that a_1b_2 intersects no other edge corresponding to a vertex of C. This means that either a_1b_2 is planar, or there is an edge e' that intersects a_1b_2 and therefore either e_1 or e_2. We keep the former case in mind and consider the latter case. W.l.o.g. assume that e' intersects e_1, the other case is symmetric. Denote by v' the corresponding vertex of e' in $\mathcal{I}$. Now, in $\mathcal{I}$, we have a path

rooted at v_1, that continues with v' and v. Denote this path by $(u_1, \ldots, u_k)$ such that $u_1 = v'$ and $u_2 = v$. Observe that $u_k = u_2 = v$ is possible.

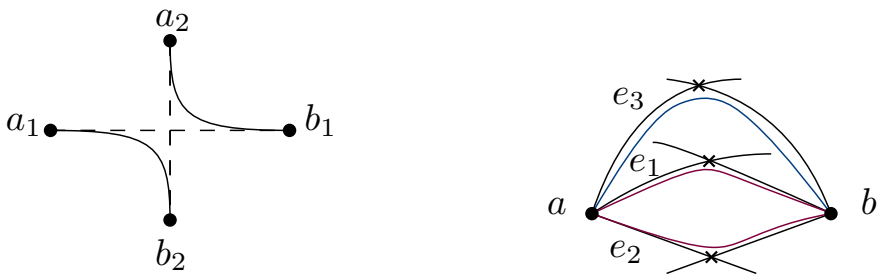


FIGURE 8.2: (A) Illustration for Lemma 22. (B) Subgraph formed by edges e_i, e'_1 for $i \leq 3$. The blue edge is a non-homotopic copy of the red ones.

By construction, the corresponding edge $a_k b_k$ of u_k in Γ has only one crossing with the corresponding edge $a_{k-1}b_{k-1}$ of u_{k-1}. Hence, if we denote by x the intersection point of these two edges, we have that the segment $a_k x$ as well as the segment $b_k x$ is crossing free in Γ. Moreover, as u_{k-1} has exactly two crossings by construction, one of the segments $a_{k-1}x$ or $b_{k-1}x$ is crossing free, w.l.o.g. assume the former. Then, there exists a (non-homotopic) planar (copy of the) edge $a_{k-1}b_k$ in Γ by maximality. Hence in both cases, we find an uncrossed edge for a fixed pair of consecutive vertices. By repeating this argumentation for any two consecutive vertices of C, we can find a set of edges S with $|S| = |C| \geq 3$ such that any edge in S is uncrossed in Γ. It remains to consider the interaction of different components of $\mathcal{I}$. Given two components X and Y of $\mathcal{I}$, it is possible that an edge ab occurs in both S_X and S_Y. We claim the following: any non-homotopic copy of an edge ab occurs in at most two such sets. Assuming we have this claim at hand, the total number of uncrossed edges in Γ is at least

$$\frac{1}{2}\sum_{i=1}^{j} X_i \geq \frac{1}{2}\sum_{i=1}^{j} 3 = \frac{3j}{2} \geq j$$

, since any cycle of G has size at least 3. Hence, there exists an assignment of the uncrossed edges of Γ (which corresponds to singletons in $\mathcal{I}$) to the critical components of $\mathcal{I}$ such that every critical component gets at least one uncrossed edge and every such edge is assigned at most once. We refer to the resulting components as *augmented* components of $\mathcal{I}$. It remains to prove the claim. Suppose for a contradiction that one copy of the edge ab belongs to at least three (critical) components X_1, X_2 and X_3 of $\mathcal{I}$. Denote by e_i and e'_i the two edges of X_i where ab is drawn along. By definition, no edge of X_i crosses an edge of X_j for $i \neq j$. But then e_1 and e'_1, e_2 and e'_2 and e_3 and e'_3 need to bound the same cell of Γ which is impossible, as e_1, e_2 and e_3 are all incident to a, while e'_1, e'_2 and e'_3 are all incident to b and we have the setting illustrated in Fig. 8.2b.

Corollary 10. *Every (augmented) component X of $\mathcal{I}$ has an independent set of size at least $\frac{|X|}{2}$.*

Theorem 22. *An n-vertex bipartite gap-planar non-homotopic multigraph has at most $4n - 8$ edges.*

Proof. Let P be the union of the maximum independent sets of every (augmented) component X of $\mathcal{I}$. Corollary 10 guarantees that $|P| \geq \frac{m}{2}$. Clearly, no two edges

of P intersect in $\mathcal{I}$, hence the edges of P induce a planar bipartite multi-graph G_P. Since G_P does not contain any homotopic-multiedges by construction, it still holds that any face of G_P has length at least four. Since Euler's formula can also be applied to non-simple graphs, we obtain Eq. (3.5) i.e., G_P has at most $2n - 4$ edges, and thus $m \leq 4n - 8$ which concludes the proof. □

In [162], the author provided a lower bound example for the class of bipartite gap-planar graphs with n vertices and $4n - 16$ edges, hence the derived upper bound is tight up to an additive constant.

Chapter 9

Fan-planarity

Introduction Another common tool to avoid clutter in dense graphs is the method of *edge-bundling*, where edges that travel in the same direction are grouped into so called *bundles*, which are usually visualized as one thicker edge [53]. Inspired by this technique, we can define the notion of *fan-planar* drawings. At a high level, fan-planar drawings have the topological property that all edges that cross a common edge have to be incident to the same vertex - moreover, these crossings have to occur "from the same side". Equivalently, one can formulate this requirement in terms of two forbidden patterns (I) and (II) shown in Fig. 9.1. In pattern (I), the edge e is crossed by two independent edges, while in pattern (II), e is crossed by two adjacent edges from different sides [1]. As we will discuss in the related work paragraph, some definitions of fan-planarity also require an additional forbidden pattern (III).

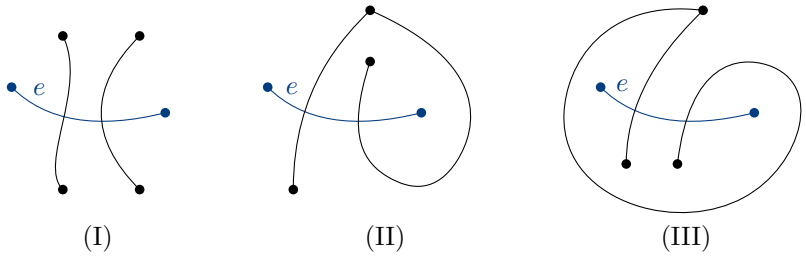


FIGURE 9.1: The three forbidden patterns in strongly fan-planar drawings.

Let us turn to a more formal definition of these patterns. Pattern (I) shows two independent edges crossing the edge e. Pattern (II) and (III) can be described as follows: Let e_1 and e_2 be the edges that cross e and let γ_1, γ_2 and γ be the corresponding Jordan arcs. By assumption, there exists $x, y, z, z' \in [0,1]$ such that $\gamma(x) = \gamma_1(z)$ and $\gamma(y) = \gamma_2(z')$. Moreover, as e_1 and e_2 are not independent, we can w.l.o.g. assume that $\gamma_1(0) = \gamma_2(1)$. Hence, the Jordan arc γ^* defined by concatenating $[\gamma_1([0,z]), \gamma([x,y]), \gamma_2([z',1]))$ is in fact a *Jordan curve*, i.e., $\gamma^*(0) = \gamma^*(1)$. By [122], curve γ^* separates the plane into two connected components such that the interior one is bounded while the exterior one is unbounded. In pattern (II), one endpoint of e lies in the interior component while in pattern (III) both endpoints of e are contained in the interior component. A drawing Γ_w is called *weakly fan-planar* if pattern (I) and (II) do not occur in Γ_w, while a drawing Γ_s is called *strongly fan-planar* if pattern (I),(II) and (III) do not occur in Γ_s[2]. In the following, we will refer to these patterns as (I), (II), and (III), respectively. Pattern (I) requires that all edges that

[1] the precise definition follows shortly

[2] We observe here that the class of fan-planar drawings is the only topological class introduced in this thesis which can only be defined in the plane and not on the sphere $\mathcal{S}^2$, as the notion of interior and exterior component is undefined in $\mathcal{S}^2$

intersect an edge e are incident to a common vertex v. In this case, v will be called the *anchor* of e.

Related work Fan-planarity was introduced in 2014 by Kaufmann and Ueckerdt [128] as drawings which do not admit (I) or (II), i.e., they considered weak fan-planarity. Consequent work used this notion of fan-planarity, an overview can be found in the recent survey by Bekos and Grilli [26].

However, it turned out that the initial work of [128], which showed that n-vertex fan-planar graphs can have at most $5n - 10$ edges, required (III) in order to allow the proof to go through - which was thus added in a subsequent journal version [129]; hence the upper bound in fact only holds for strong fan-planar graphs. For all intermediate work on fan-planarity, one has to carefully read the assumptions on whether weak or strong fan-planarity was considered. In particular, the result of Brandenburg [45] is influenced by this correction. He studied a version of fan-planarity where only (I) is forbidden - which he refers to as *adjacency crossings* graphs. He establishes that weak fan-planar graphs form a proper subset of adjacency crossing graphs which motivates the existence of (II)[3]. Further, he "shows" that n-vertex adjacency-crossing graphs have at most $5n - 10$ edges. This result is obtained by a construction which transforms any adjacency-crossing graph into a weak fan-planar graph with the same number of edges. And here is the issue - at the time of publication, the upper bound of $5n - 10$ edges was assumed to hold for weak fan-planar graphs - but in fact, we do not have an upper bound on the edge-density of weak fan-planar graphs to date.

Our Contribution We show in Section 9.1 that pattern (III) is not redundant. Namely, we show that there exists graphs which are weakly fan-planar but not strongly fan-planar. On the other hand, we will show in Section 9.2 that the density results for (bipartite) strongly fan-planar graphs derived in ([10]) [129] also hold for (bipartite) weakly fan-planar graphs. Our result ensures that the upper bound for the edge density of adjacency-crossing graphs derived in [45] is in fact correct. In Section 9.3 we will establish that (bipartite) strongly fan-planar graphs have thickness at most (two) three by a complete characterization of odd cycles that can occur in the intersection graph of any fan-planar drawing.
Section 9.1 and Section 9.2 is based on joint work with Otfried Cheong, Henry Förster, Julia Katheder and Lena Schlipf which appeared at the "31st International Symposium on Graph Drawing". Section 9.3 is based on joint work with Otfried Cheong and Lena Schlipf which was published at the "30th International Symposium on Graph Drawing" [54].

9.1 Weak vs strong fan-planarity

In this section, we will establish that strongly fan-planar graphs form a proper subset of weak fan-planar graphs.

Theorem 23. *There exists a weakly fan-planar graph that does not admit a strongly fan-planar drawing.*

Proof. Let Γ be a weakly fan-planar drawing of graph G which does not admit a strong fan-planar drawing. Clearly, any (weakly) fan-planar drawing, and hence Γ,

[3] obviously, (I) is a necessity to define fan-planartiy, as otherwise every geometric drawing is feasible

contains at least one forbidden pattern (III). In order to guarantee the existence of at least one pattern (III) in any valid fan-planar drawing of G, we will use the following key idea. We start with a planar graph with a unique embedding. We will then make every edge of this planar graph "uncrossable" by replacing it with a suitable gadget introduced by [35]. Afterwards, we insert into every face of the planar graph a small gadget graph (shown in Fig. 9.2a) which can only be drawn if we allow (III). In order to achieve our goal, we will leverage the following lemma.

Lemma 23 ([35]). *Let $\mathcal{P}$ be the planarization of any (weakly) fan-planar drawing of K_7. Then, between any pair of (real) vertices of $\mathcal{P}$, there exists a path which does not contain a real edge of G.*

We are now ready for the detailed version of the proof of Theorem 23. Let G_q be a 3-connected planar quadrangulation (e.g., one that is obtained by the construction in [47]). Note that by construction, G_q is bipartite and G_q has a unique embedding in S^2 [182]. In the next step, we insert a copy of our gadget graph H shown in Fig. 9.2a into every face f of G_q by identifying the outer cycle of H with the facial cycle of f. Denote by G_q^+ this supergraph of G_q. We use the color scheme of Fig. 9.2a to color all edges of G_q^+ - in particular, the edges of G_q form a subset of the red edges of G_q^+. In the next step, we will substitute every red edge of G_q^+ with a K_7 and denote the resulting graph by G_s. We claim that G_s is weakly fan-planar, but not strongly fan-planar. The first statement is true since K_7 admits a fan-planar drawing, see Fig. 9.2b, while Fig. 9.2a shows a weakly fan-planar drawing of gadget graph H - thus G_s admits a weakly fan-planar drawing. Consider now the second statement. Let $\mathcal{G}$ be the set of all tuples (G_s, Γ) where Γ is a weakly fan-planar drawing of G_s. Let $\mathcal{G}' \subseteq \mathcal{G}$ be the set of tuples (G_s, Γ) where Γ contains the least number of patterns (III) and finally let $\mathcal{G}'' \subseteq \mathcal{G}'$ be the set of tuples (G_s, Γ) where Γ contains the least number of crossings. In the remainder, we fix such a tuple $(G_s, \Gamma) \in \mathcal{G}''$. We will prove that Γ contains at least one pattern (III) and hence **every** weakly fan-planar drawing of G_s requires at least one pattern (III) by our choice of Γ. Consider a red edge $ab \in G_q^+$ and denote by $a = v_1, \ldots, v_7 = b$ the vertices of the K_7 which substitute ab in G_s. By Lemma 23, there exists a sequence of crossed edges S from a to b. By (I), no edge which is not incident to one of $v_1, \ldots, v_7$ can intersect S. By construction, the only edges incident to vertices $v_2, \ldots, v_6$ are the ones contained in the K_7. Hence, the only edges that potentially intersect S and interact with the remainder of G_s are incident to either $a = v_1$ or $b = v_7$. Suppose for a contradiction that there exists an edge incident to a or b that crosses S in Γ such that its other endpoint does not belong to the vertices of the K_7. But then we can easily reroute the edge such that its crossing with S can be avoided, see Fig. 9.2c, and we obtain a contradiction to our choice of Γ. Hence, with a slight abuse of notation, we conclude that the red edges are *uncrossed* in Γ.

Recall that G_q has a unique embedding in S^2, and hence a unique embedding in the plane after we fix the outer-face. Since H is inserted into every face of G_q, it is ensured that for some face f of G_q, the four vertices that bound f, which were identified with the outer cycle of our gadget graph H, indeed form a facial walk that contains H in its interior. Denote by u and u' the two vertices of f that are of the same partition. By 3-connectivity, the only face in G_q where both u and u' are part of a facial cycle is f. Since the red edges are uncrossed, it follows that for any green/blue edge contained in the gadget, both endpoints belong to the same face of the subgraph induced by the red edges. In particular, this implies that vertices v, v', w, w' and z, refer to Fig. 9.2a, lie on a common face of the red subgraph. Further, the edge (u, z') implies that the gadget can not be mirrored on the horizontal axis,

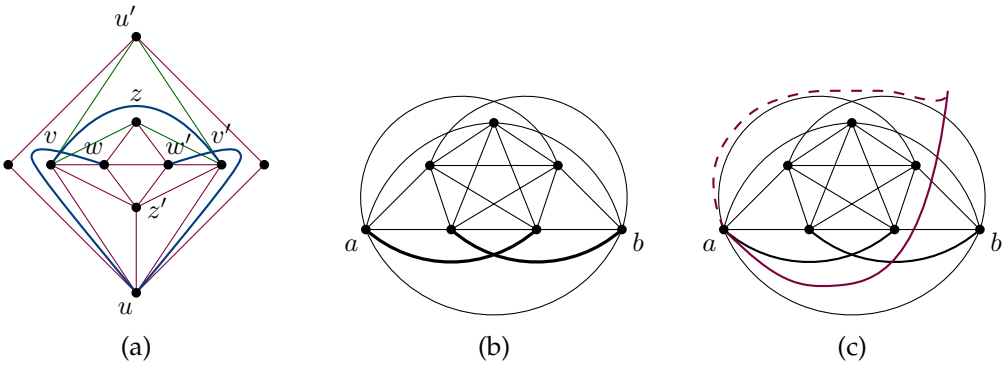


FIGURE 9.2: (A) Gadget graph H. (B) A fan-planar drawing of K_7, the fat edges form a path from a to b that contains no uncrossed edge. (C) An edge incident to a that crosses the K_7 can be rerouted in order to avoid this crossing.

as otherwise the red edge (u, z') would intersect either (u', v) or (u', v'). Hence, the shown embedding is unique (up to a mirroring on the vertical axis, which is symmetric). Now it is easy to verify that the edges (u, w), (u, w') and (v, v') always form (III), which concludes the proof. □

9.2 Density of weakly fan-planar graphs

In this section, we will show that the density results for strongly fan-planar graphs also transfer to the weakly fan-planar one. Let us call a triple e, e_1, e_2 of edges in a weakly fan-planar drawing a *heart* if e_1 and e_2 share an endpoint u, both cross e such that they form pattern (III), and the part of e between the crossings with e_1 and e_2 is not crossed by any edge of the graph, see Fig. 9.3a. We start with the following observation:

Lemma 24. *Let Γ be a weakly fan-planar drawing that is not strongly fan-planar. Then Γ contains a heart.*

Proof. By assumption, Γ contains three edges e, e_1, e_2 that form pattern (III), where e_1 and e_2 share endpoint u and cross e. Let E' be the set of edges that cross e. By (I) and (II), any edge $e' \in E'$ must be incident to u, and by (II) it must cross e from the same side as e_1 and e_2. The edges of E' cannot cross each other since they share an endpoint, and each edge $e' \in E'$ forms pattern (III) either with e and e_2, or with e and e_1. Let $E_1 \subset E'$ be the set of edges of the first kind, $E_2 = E' \setminus E_1$ the second kind. If we order E' by their crossing point with e along e, then we first encounter all elements of E_1, then all elements of E_2. Picking the last element of E_1 and the first element of E_2 together with e is the heart we are looking for. □

We will call the sets E_1 and E_2 as defined in the previous proof *left valve* and *right valve* of a heart. We also call a weakly fan-planar drawing Γ of a graph G *minimal* if it contains the smallest possible number of triples of edges that form pattern (III).

The following lemma provides an important observation for minimal weakly fan-planar drawings.

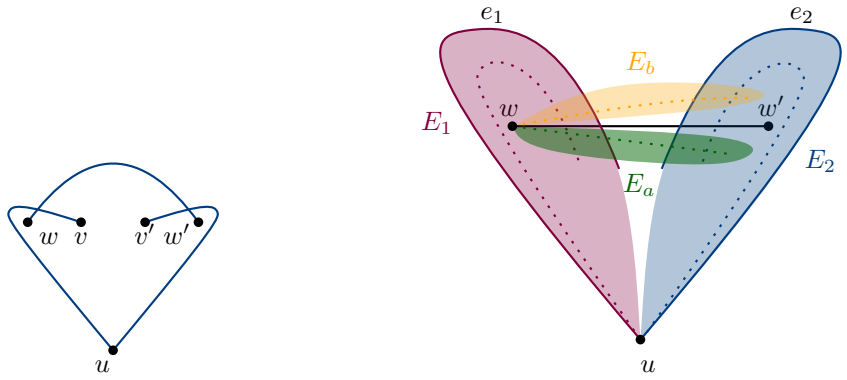


FIGURE 9.3: (A) A heart. (B) Illustration of the setting of Lemma 25.

Lemma 25. *Let e, e_1, e_2 be a heart in a minimal weakly fan-planar drawing Γ. Then there is no edge $e' \neq e$ in Γ that crosses both e_1 and e_2.*

Proof. The proof of Lemma 25 is quite technical and will cover several pages; hence, after introducing some required notation, we will give a brief outline of the proof before diving into the technical details. Let u be the common endpoint of e_1 and e_2, let $e = (w, w')$, and assume there exists an edge $e' \neq e$ that intersects both e_1 and e_2. Denote by x_1 and x_2 the intersection point of e_1 and e and e_2 and e, respectively. By (I), e and e' share an endpoint, say w. This implies by (I) and (II) that *every* edge which crosses e_1 or e_2 is incident to w. W.l.o.g. assume that x_1 is encountered before x_2 when traversing e starting at w. We will define four sets of edges that will be helpful for the remainder of the proof. In particular, we construct these sets based on the edge routing in the original drawing Γ - while some of the edges may be redrawn at a later time, they will belong to their initially assigned set throughout the proof. Let E_1 and E_2 be the sets of edges that correspond to the left valve and the right valve of the heart induced by e_1, e_2, e - in particular, $e_1 \in E_1$ and $e_2 \in E_2$. Further, let E_b be the set of edges which cross both e_1 and e_2 *before* e, that is, for an edge $e_b \in E_b$, the intersection point of e_1 and e_b (e_2 and e_b) is encountered before x_1 (before x_2), when traversing e_1 (e_2) starting at u. Symmetrically, let E_a be the set of edges which cross both e_1 and e_2 *after* e. Note that by (I) and (II), any edge that intersects both e_1 and e_2 is either part of $E_b \cup E_a$ or coincides with e. Further, all edges of $E_b \cup E_a$ are incident to w by (I). Finally, we observe that the existence of e' ensures that at least one of E_a or E_b is nonempty. For an illustration, see Fig. 9.3b.

Outline Let us layout the rough idea of the proof before we dive into the technical details. In the first step, we will "flip" all edges of E_1 to the other side as illustrated in Fig. 9.4 to obtain the intermediate drawing Γ'. The crucial observation is that Γ' does not contain any pattern (III) introduced by two edges of $E_1 \cup E_2$ and an edge of $E_a + \{e\}$, while Γ contains at least one such pattern (induced by e_1, e_2 and e). Unfortunately, we are not done after this initial step, as the flip-operation may induce two types of pattern (III) in Γ' which were not present in Γ - one that consists of an edge of E_a, an edge of E_b and an edge e'' incident to u which does not cross e (and hence belongs to neither E_1 nor E_2), see Fig. 9.5, while the other one consists of an edge of E_b, an edge of E_a and an edge of E_1, see Fig. 9.6 for an illustration.

We will solve both these issues by rerouting the edges of E_b as shown in Fig. 9.7 which will yield the final drawing Γ''.

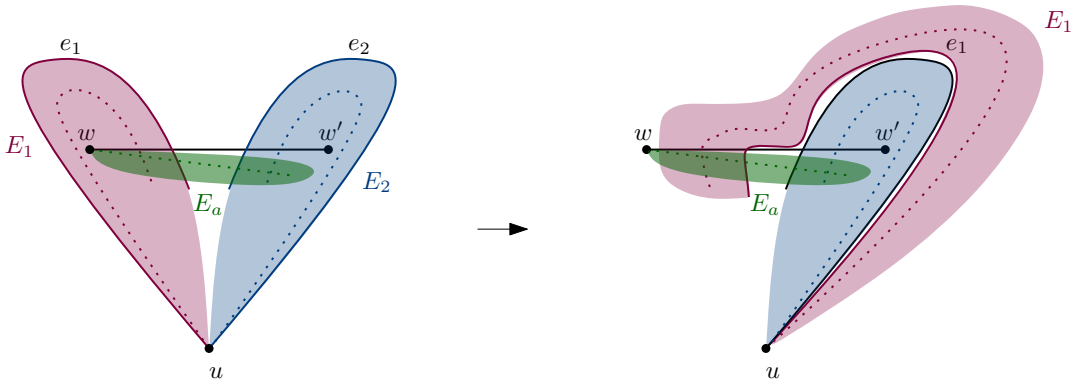


FIGURE 9.4: Transformation from Γ to Γ'

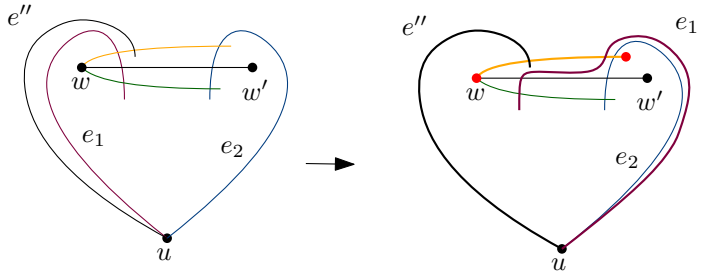


FIGURE 9.5: Fat lines form (III) which completely contains the red endpoints in its interior

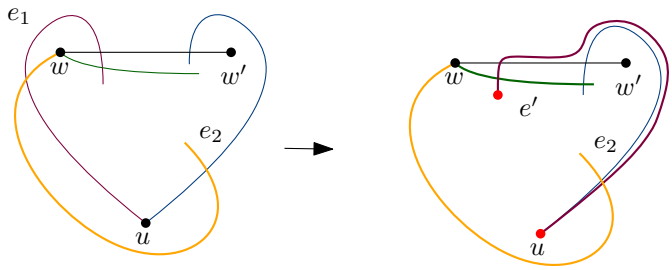


FIGURE 9.6: Fat lines form (III) which completely contains the red endpoints in its interior

Flipping the edges of E_1 Refer to Fig. 9.4 for a visualization. Let $e_1^1, e_2^1, \dots e_k^1$ be the edges of E_1 in the order that they intersect edge e in Γ starting at w, which implies that e_k^1 coincides with e_1. We consider the edges in reverse order and start with $e_k^1 = e_1$. The curve of e_k^1 in Γ' follows the curve of e_2 slightly outside until x_2, then it follows e until x_1, where the curve intersects e and afterwards it inherits its original curve in Γ. The curve of e_{k-1}^1 again follows the curve of e_2 (slightly outside the new curve of $e_k^1 = e_1$) until x_2, where it follows e until the intersection point of e_{k-1}^1 with e in Γ. Here, the curve intersects e and then again inherits its original curve in Γ until it reaches its endpoint different from u. We repeat this procedure for all edges of E_1 to obtain the intermediate drawing Γ'.

Γ' is weakly fan-planar Let us first assert that Γ' is weakly fan-planar. Fix an edge $e_i^1 \in E_1$ and consider its curve γ in Γ'. Curve γ consists of three parts - the first part follows e_2 starting at u until x_2, the second part follows e until the intersection point of e_i^1 and e in Γ and the last part is inherited from Γ. Clearly, any additional crossing

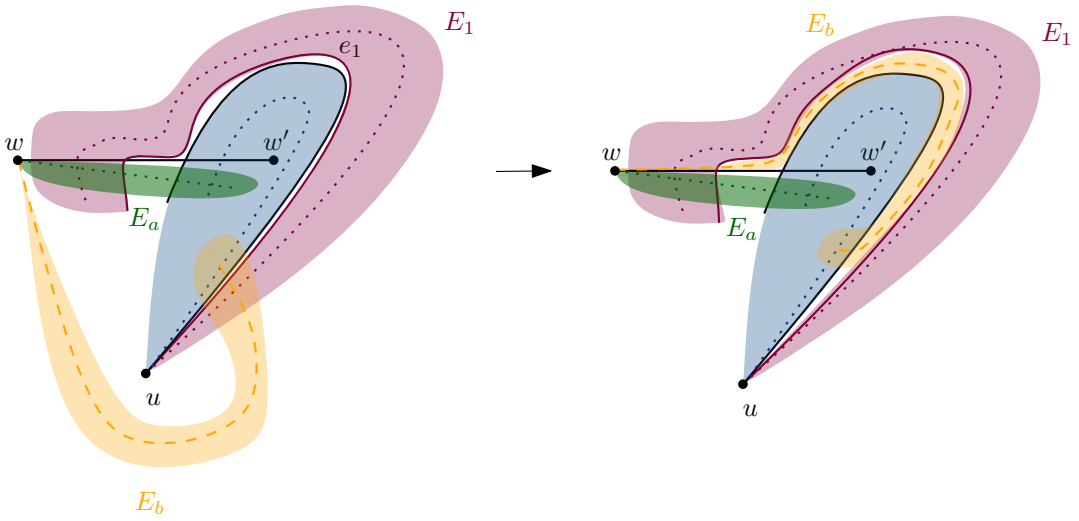


FIGURE 9.7: Transformation from Γ' to Γ''

that we introduce may only occur on the first or second part of γ. By construction, however, the second part is crossing free, as the segment of e between x_1 and x_2 in Γ is crossing-free since e_1, e_2 and e form a heart; and since we consider the edges in the reverse order that they intersect e (starting at w), they do not intersect each other. Let us now consider the first part of γ. By construction, any edge that intersects this first part in Γ' also intersects e_2 and is therefore incident to w by (I).

Thus, if we manage to show that any crossings of the third part of γ are due to edges which are incident to w, we neither introduce pattern (I) nor (II) by the flip-operation, as the latter would require a nonempty region between two first-parts of such curves, which is a contradiction to our construction. To prove the claim, consider the closed region $\mathcal{R}$ defined by w, x_1 and the intersection of e_1 and e' in the original drawing Γ. Suppose first that the intersection with e is encountered before the intersection with e' when traversing e_1 starting at u, see Fig. 9.8 for an illustration of this case.

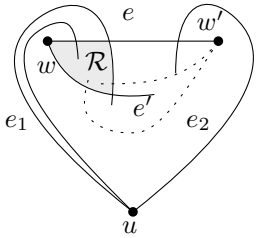


FIGURE 9.8: Any edge of E_1 that enters region $\mathcal{R}$ has w as its anchor.

By definition, every edge in E_1 besides e_1 enters $\mathcal{R}$ over e. If such an edge is leaving $\mathcal{R}$, then it has to also cross e' by simplicity, but then its anchor is w and by (I) it can only be crossed by edges that are incident to w. Suppose now an edge of E_1 ends in $\mathcal{R}$, but its anchor is not w. Since the edge intersects e, its anchor is therefore w'. But no edge incident to w' can enter $\mathcal{R}$, since it would either cross e_1, whose anchor is w and hence it would coincide with $e = (w, w')$, it cannot cross e by simplicity and if it crosses e', then it has to be incident to the anchor of e', which is u, but then the curve has to intersect the boundary of $\mathcal{R}$ twice which is impossible.

Suppose now that the intersection of e with e' is encountered before e. All edges of E_1 cross e by definition. Further, no two edges of E_1 can intersect each other

by simplicity. Hence, since the edges of E_1 intersect e on the segment that bounds $\mathcal{R}$, it follows that they also intersect e', and hence their anchor is w. Thus, we have established that Γ' is indeed weakly fan-planar and we can now consider the number of patterns (III).

Number of patterns (III) Clearly, in any new pattern (III) in Γ', at least one edge of E_1 is involved. Assume that $e_i \in E_1$ is involved in a new pattern (III) which is not present in Γ. Suppose first that two edges incident to u are involved in this new pattern. Any such pattern contains an edge e^* incident to w and an edge $e'' \notin E_1$ incident to u which was not flipped, i.e., its first part follows the first part of the curve of e_1 in Γ. Assume that $e^* \in E_a$. In order for e'' to cross e^*, it also has to cross e and hence belongs to E_1, a contradiction. Hence, $e^* \in E_b$ has to hold. We refer to this pattern as *type-one*, which is illustrated in Fig. 9.5. Suppose now that two edges incident to w and one edge of E_1 form a new pattern (III). We observe that the incident edges to w cannot both belong to E_b or E_a - hence any such pattern consists of one edge of E_a, one edge of E_b and one edge of E_1, which we call *type-two*, which is illustrated in Fig. 9.6.

Rerouting the edges of E_b We will construct a drawing Γ'' that avoids all patterns of type-one or type-two by rerouting the edges of E_b as shown in Fig. 9.7. Let $e_1^b, e_2^b, \dots e_k^b$ be the edges of E_b in the order that they intersect edge e_2 in Γ starting at u. We consider the edges in reverse order and start with e_k^b. The curve of e_k^b in Γ' follows the curve of e slightly outside until x_2, then it follows e_2 until the intersection point of e_k^b with e_2 in Γ, and then it inherits its original curve in Γ. Similar to the flip operation of E_1, we repeat this procedure for all edges of E_b to obtain our final drawing Γ''. Clearly, no two edges of E_b cross each other in Γ''. Consider an edge $e_i^b \in E_b$ with curve γ_b in Γ''. The first part of γ_b only crosses edges which also cross the edge e - as e is not involved in any pattern (III) in Γ', neither is the first part of γ_b. Recall that type-one patterns required our edge e_i^b to intersect an edge e'' which was not part of E_1, i.e., which did not intersect e. Hence, γ does not intersect e'' in Γ' and thus we resolve all type-one patterns. Moreover, e and and the first part of γ_b cross all these edges from the same side - hence, (I) and (II) are maintained so far. Consider now the second part of γ_b which follows e_2 starting at x_2 until the intersection point of e_i^b and e_2 in Γ and suppose there is an edge that crosses it. By construction, this edge also crosses e_2 and thus is incident to w by (I). Moreover, this edge was already present in Γ (as no edge of E_1 nor E_b, which are the only edges that differ from Γ to Γ'', actually cross the second part of γ_b) and has to intersect e_2 from the same side as e_i^b in Γ to not violate (II)- but then this edge necessarily belongs to E_b and thus does not cross e_i^b in Γ''. In particular, this resolves all patterns of type-two. Thus, both the first and the second part of γ_b are crossing free - since the third part of γ_b is inherited from Γ, we do not introduce additional crossings and hence no edge of E_b is involved in a pattern (III) in Γ''. To conclude, Γ'' is a weakly fan-planar drawing whose set of patterns (III) forms a proper subset of the one of Γ, since the triplet e_1, e_2 and e does not induce (III) in Γ'' and we obtain a contradiction. □

Theorem 24. *A weakly fan-planar graph G with n vertices has at most $5n - 10$ edges.*

Proof. We proceed by induction on the number of edge triples forming pattern (III) in a *minimal* weakly fan-planar drawing Γ of graph G. In the base case, this number is zero, so Γ is strongly fan-planar. Then G is strongly fan-planar, and has at most $5n - 10$ edges by [129]. For the induction step, we have by Lemma 24 that Γ contains

a heart $\mathcal{H}$ formed by the edges e, e_1 and e_2, see Fig. 9.3a. Let x_i be the crossing point of e_i and e, for $i = 1, 2$, and denote by $\mathcal{L}$ the closed curve that consists of the partial curves of the edges e_1, e_2 and e up to the crossing points x_1 and x_2. Denote by G_1 the subgraph of G consisting of those vertices and edges of G that lie (entirely) in the *closed* region bounded by $\mathcal{L}$. In particular, the vertices u, w, v, v', and w', and the edges e_1,e_2, and e all belong to G_1. Similarly, let G_2 be the subgraph of G consisting of those vertices and edges of G that lie entirely in the *unbounded closed* region defined by $\mathcal{L}$. In particular, vertex $u \in G_2$, but none of the edges e, e_1, e_2 is in G_2. Let $|V[G_2]| = r$ and thus $|V[G_1]| = n - (r - 1)$, as u is part of both G_1 and G_2.

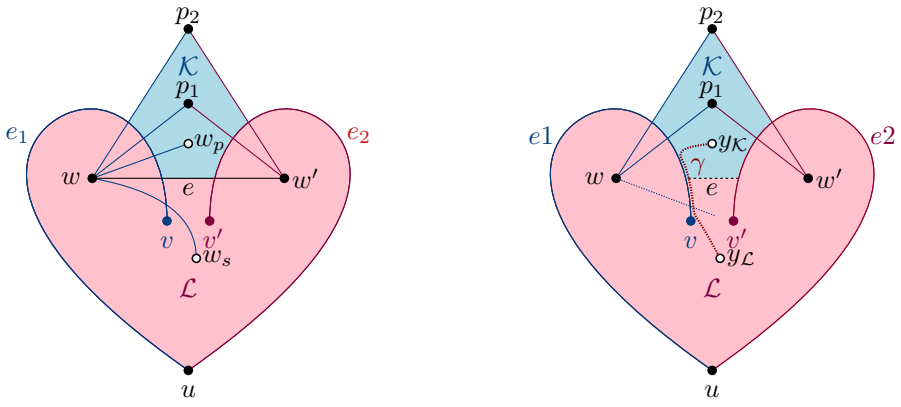
Edges that are in neither G_1 or G_2. Denote by E_{ext} the set of edges that belong to neither G_1 nor G_2 and consider an edge $e' \in E_{ext}$. Clearly, e' must cross $\mathcal{L}$. This crossing cannot be on e by the heart property, so it must be on e_1 or e_2. The edge e' cannot cross *both* e_1 and e_2 by Lemma 25, so e' either crosses e_1 and must be incident to w or crosses e_2 and must then be incident to w' by (I). We claim that e' can only cross edges incident to u outside of $\mathcal{L}$. To see this, suppose for a contradiction that e' is crossed by an edge e'' outside of $\mathcal{L}$ which is not incident to u. W.l.o.g. assume that e' is incident to w and crosses e_1, the other case is symmetric. By (I), edge e'' is then necessarily incident to v, which is contained inside $\mathcal{L}$. Since the intersection point x of e' and e'' lies outside of $\mathcal{L}$ by assumption, if we follow the curve of e'' starting at v until x we necessarily have to exit $\mathcal{L}$, i.e., intersect either e, e_1 or e_2. If e'' would intersect e, then it is necessarily also incident to u by (I), a contradiction to our assumption. Further, e'' cannot intersect e_1 by simplicity as they share the common vertex v. If e'' intersects e_2, then by (I) its other endpoint (different from v) is v' which is contained inside $\mathcal{L}$ as well. This implies that e'' has to leave and reenter $\mathcal{L}$ - but as previously observed, it cannot cross either e or e_1, so it would cross e_2 twice, which is a contradiction to simplicity.

In the following, we will distinguish two cases based on the number of common neighbors between w and w' (restricted to the edges of E_{ext}).

w and w' have at least two common neighbors. In this case, we will construct a (weakly) fan-planar graph G' from G with the same number of vertices and edges as G such that G' contains strictly less patterns (III) then G, and hence by induction the desired result holds. This will be achieved by exchanging the edge e of $\mathcal{H}$ with a suitable edge $(y_\mathcal{K}, y_\mathcal{L}) \notin E$, where $y_\mathcal{K}$ and $y_\mathcal{L}$ will be defined in the following. To this end, assume w.l.o.g. that $\mathcal{H}$ is chosen so that $\mathcal{L}$ does not completely contain any other heart $\mathcal{H}'$. We aim to find a sequence of curve-segments in Γ, along which we can insert the edge $(y_\mathcal{K}, y_\mathcal{L})$ without introducing (I), (II) or (III). We will refer to this constructed curve as γ. Refer to Fig. 9.9a for the following case analysis.

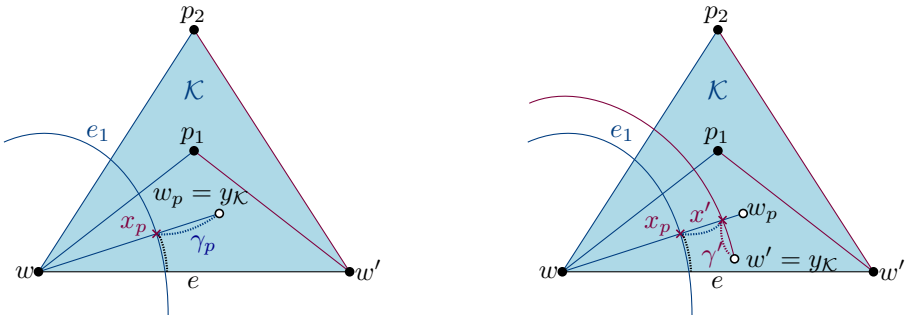
Denote by p_1 and p_2 two common neighbors of w and w'. Since the edges of w and w' to p_1 and p_2 both cross edges incident to u (as they exit $\mathcal{L}$) they cannot cross each other. Hence there exists a proper nesting of p_1 and p_2, w.l.o.g. assume that p_1 is contained in the region delimited by e,e_1,e_2,(w, p_2) and (w', p_2), which will be called $\mathcal{K}$. Further, denote by w_p and w_s the *predecessor* and *successor* of the edge e at w restricted to the edges incident to w that cross e_1, i.e., the edge $e_p = (w, w_p)$ crosses e_1 immediately before e and $e_s = (w, w_s)$ immediately after e when following e_1 starting at u. Note that, it is possible that w_p coincides with p_1 and w_s does not necessarily exist.

We first identify a suitable vertex $y_\mathcal{K}$ and a crossing-free subcurve of γ that connects $y_\mathcal{K}$ with the segment of e between x_1 and x_2. Consider w_p. By Lemma 25,



(a) (b)

FIGURE 9.9: (a) Regions $\mathcal{K}$ and $\mathcal{L}$ and vertices w_p and w_s. (b) Replacement of e.



(a) (b)

FIGURE 9.10: Identification of $y_\mathcal{K}$.

(w, w_p) cannot cross both e_1 and e_2, hence w_p lies outside of $\mathcal{L}$ [4]. Further, since edges of w and w' do not intersect, w_p is necessarily contained inside $\mathcal{K}$. Let x_p be the crossing point of e_p and e_1 in Γ. Observe that as e_p is the predecessor of e, the curve-segment of e_1 between x_1 and x_p is crossing-free, hence we add it to γ. To identify the endpoint y, consider the curve-segment γ_p of e_p between x_p and w_p.

First, assume that γ_p is crossing-free. In this case, we identify $y_\mathcal{K}$ with w_p and append γ_p to γ; see Fig. 9.10a.

Second, assume that γ_p has at least one crossing. Let e' be the first edge crossing e_p when traversing γ_p from x_p. Let x' be the intersection of e' and e_p. Observe that the segment from x' to x_p on e_p is crossing-free and hence can be appended to γ. Since (x_1, x_2) is crossing-free by the heart property, the anchor of e_p must be u. Hence, e' also intersects the edge (w, p_2), its anchor is w, and its endpoint w' lies inside $\mathcal{K}$ (its other endpoint is u). Finally, we observe that the curve γ' of e' between x' and w' is crossing free, since otherwise (w, w_p) is not the predecessor of e at w. Thus, we fix w' as $y_\mathcal{K}$ and append γ' to γ; see Fig. 9.10b.

Observe that in both cases $y_\mathcal{K}$ lies in $\mathcal{K}$. Also note that $y_\mathcal{K}$ can only be adjacent to w and w' inside $\mathcal{L}$ as the segment of e between x_1 and x_2 is uncrossed whereas w and w' are the anchors of e_1 and e_2, respectively. Hence, we can be sure that $(y_\mathcal{K}, y_\mathcal{L})$ does not exist in G yet as long as $y_\mathcal{L} \notin \{w, w', u\}$ and $y_\mathcal{L}$ lies in $\mathcal{L}$; see Fig. 9.9b. Thus,

[4] by simplicity, it can neither cross e nor e_1 a second time

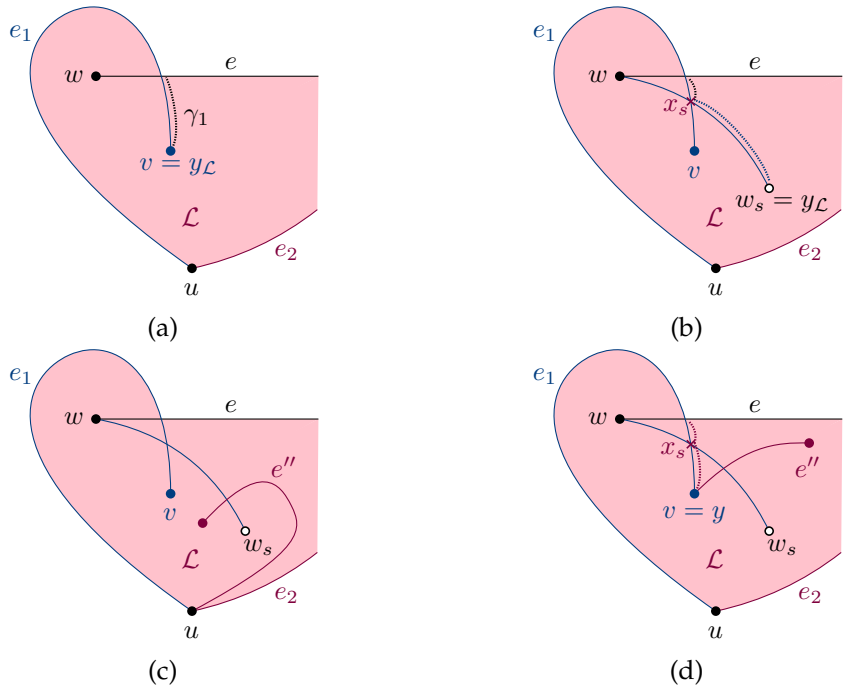


FIGURE 9.11: Identification of $y_{\mathcal{L}}$.

it remains to extend γ starting from its intersection with e to a suitable vertex $y_{\mathcal{L}}$ that is contained in $\mathcal{L}$.

Let v be the endpoint of e_1 different from u. Consider the curve γ_1 of e_1 from x_1 to v. First, if γ_1 is crossing-free, we identify $y_{\mathcal{L}}$ with v and add γ_1 to γ; see Fig. 9.11a.

Otherwise, γ_1 contains a crossing. Assume momentarily that γ_1 is crossed by an edge incident to w'. In this case, we switch w and w' and observe that this case cannot occur for w' at the same time by (I). Thus, we conclude that e_s must be present. Denote by x_s the crossing between e_1 and e_s. We add the curve segment between x_1 and x_s to γ. If the curve of e_s between w_s and x_s is crossing-free, we add it to γ and identify w_s with $y_{\mathcal{L}}$; see Fig. 9.11b.

Note that in all cases discussed so far, γ is crossing-free when removing e. Thus, when replacing e with $(y_{\mathcal{K}}, y_{\mathcal{L}})$, the number of (I) stays at most the same while the number of (III) decreases by at least 1; see Fig. 9.9b (in this case the dotted edge at w does not exist).

Finally, it remains to consider the case where there is an edge e'' crossing e_s. We choose e'' such that it is the first edge crossing e_s after x_s when traversing e_s from x_s to w_s. Note that the anchor of e_s is either v or u. If it is u, we observe that e_1, e_s and e'' form a heart inside $\mathcal{L}$ (see Fig. 9.11c); a contradiction to the choice of $\mathcal{H}$. Thus, the anchor of e_s must be v. In this case, we again choose $y_{\mathcal{L}} = v$ and append the segment between x_s and v to γ. In this case γ will cross (w, w_s). However, the anchor of (w, w_s) is $y_{\mathcal{L}} = v$ so the number of (III) is maintained. Moreover, since γ intersects (w, w_s) in between e_1 and e'' the number of (III) still decreases by at least 1 when replacing e with $(y_{\mathcal{K}}, y_{\mathcal{L}})$; see Fig. 9.9b, which concludes our description of G'. Now, induction easily yields the desired result.

w and w' have at most one common neighbor. Let us now consider the case where w and w' share at most one common neighbor. Recall that by definition, Γ restricted to G_2 is weakly fan-planar and does not contain the pattern (III) formed by $\mathcal{H}$, hence

the inductive assumption applies and we have

$$|E(G_2)| \leq 5|V(G_2)| - 10.$$

More work is required for G_1. We need to create a new weakly fan-planar drawing Γ' for G_1 that has at least one fewer edge triple forming pattern (III). Let E_1 and E_2 be the left valve and the right valve of our heart, respectively, such that $e_1 \in E_1$ and $e_2 \in E_2$ holds. Recall that G_1 lies entirely in the bounded region $\mathcal{L}$ defined by the partial curves of e_1, e_2 and e up to their respective intersection points. In particular, this implies that the partial curve of e_1 between u and x_1 is crossing free. Moreover, the partial curve of e between x_1 and x_2 is also crossing free as e_1, e_2 and e form a heart. We will use these two observations to "flip" all edges of E_1 as follows. The attentive reader may already guess that this flip-operation is equivalent to the one performed in the proof of Lemma 25, hence refer to Fig. 9.4 for an illustration.

Flipping the edges of E_1. Let $e_1^1, e_2^1, \dots e_k^1$ be the edges of E_1 in the order that they intersect edge e in Γ starting at w, which implies that e_k^1 coincides with e_1. We consider the edges in reverse order and start with $e_k^1 = e_1$. The curve of e_k^1 in Γ' follows the curve of e_2 slightly outside until x_2, then it follows e until x_1, where the curve intersects e and afterwards it inherits its original curve in Γ. The curve of e_{k-1}^1 again follows the curve of e_2 (slightly outside the new curve of $e_k^1 = e_1$) until x_2, where it follows e until the intersection point of e_{k-1}^1 with e in Γ. Here, the curve intersects e and then again inherits its original curve in Γ until it reaches its endpoint different from u. We repeat this procedure for all edges of E_1 to obtain drawing Γ'.

Γ' is weakly fan-planar and contains less pattern (III). To show that Γ' is weakly-fan-planar and contains at least one pattern (III) less than the drawing of G_1 in Γ, fix an edge $e_i^1 \in E_1$ and let γ be its curve in Γ'. Observe that by construction, the part of γ starting at u until the intersection point of e_i^1 with e in Γ is crossing free in Γ', as no two edges of E_1 intersect in Γ' by construction and any other edge that would intersect γ would also cross either e_2 or e in Γ and thus be contained both inside and outside $\mathcal{L}$. Hence, all crossings of $e_i^1 \in E_1$ are on the part of γ which was inherited from Γ; it follows that (I) is maintained in Γ'.

Suppose now that e_i^1 is part of a pattern (II) or (III) in Γ', which was not present in Γ. Suppose first that only one edge of this pattern is incident to u, i.e., e_i^1 uniquely defines this edge. Recall that the only part of γ that can be involved in crossings is the one inherited from the curve of e_i^1 in Γ - but then clearly the pattern was already present in Γ, a contradiction.

Suppose now that two edges of this pattern are incident to u. Again, since the only part of the curve of e_i^i which is different from Γ' to Γ is uncrossed, the existence of (II) in Γ' would imply the existence of (II) in Γ, a contradiction since Γ is weakly fan-planar. It remains to consider the case where two edges e_x and e_y that are incident to u form (III) together with an edge e'. Clearly, at least one of e_x and e_y belongs to E_1, as otherwise the pattern was already present in Γ. First observe that by construction we have $e' \neq e$. As there is no edge in G_1 which crosses an edge of E_1 and an edge of E_2 besides e by Lemma 25, such a pattern could only occur between two edges of E_1. But as this pattern was not present in Γ by assumption and since the closed region defined by the partial curves of the edges until their intersection points with e contains no endpoint, and thus in particular no endpoint of e', it

follows that they do not form (III) in Γ'. Hence, our new weakly fan-planar drawing Γ' has $|E_1| \times |E_2| \geq 1$ patterns (III) less than Γ and we can apply the inductive assumption to get

$$|E(G_1)| \leq 5|V(G_1)| - 10.$$

It remains to consider the edges of E_{ext}. Assume that $|E_{ext}| = k$. By assumption, the edges of E_{ext} are incident to at least $k-1$ vertices outside of $\mathcal{L}$. To make up for these edges, we construct a new graph G_2' from G_2 as follows. Recall that the drawing $\Gamma[G_2]$ derived from Γ contains an empty region inside $\mathcal{L}$. We insert a single vertex v^* inside this region and connect it to the (at least) $k-1$ neighbors of w and w' as well as to u. Recall that any edge of E_{ext} only crossed edges incident to u outside of $\mathcal{L}$ and therefore (I) and (II) is maintained. Moreover, since the new edges are routed along the curves of the removed ones, any pattern (III) was necessarily already present in Γ. Hence, we augmented G_2 to the weakly fan-planar graph G_2' with $r+1$ vertices and at least one pattern (II) less than G (the one formed by e,e_1 and e_2) and hence we can apply the induction hypothesis:

$$\begin{aligned}
|E[G]| &\leq |E(G_1)| + |E(G_2)| + k \\
&\leq |E(G_1)| + (|E(G_2')| - k) + k \\
&\leq 5(n-r+1) - 10 + 5(r+1) - 10 \\
&= 5n - 10,
\end{aligned}$$

which concludes the proof. □

For bipartite graphs, we proceed in a similar way.

Theorem 25. *An n-vertex bipartite weakly fan-planar graph has at most $4n-12$ edges.*

Proof. Let Γ be a weakly-fan planar drawing of G with the least number of forbidden patterns (III). Denote this number by i. We proceed by induction on i. For the base case, we have that $i = 0$, but then Γ is also strongly fan-planar and hence G has at most $4n-12$ edges by [10]. For the induction step, we proceed as in the second case of the proof of Theorem 24 with two main differences. Instead of inserting a single vertex v^*, we have to insert two vertices v_a and v_b in order to guarantee bipartiteness. Note that in the bipartite case, we have exactly k neighbors (and not possibly less than that) of w and w' in G_2, as a common neighbor among w and w' would imply a 3-cycle. Further, we can add the edge (v_a, v_b) as well as one of (u, v_a) or (u, v_b). In total, we get

$$\begin{aligned}
|E[G]| &\leq |E(G_1)| + |E(G_2)| + k \\
&\leq |E(G_1)| + (|E(G_2')| - (k+2)) + k \\
&\leq 4(n-r+1) - 12 + 4(r+2) - 12 - 2 \\
&\leq 4n - 12,
\end{aligned}$$

which concludes the proof.

□

9.3 Thickness of strongly fan-planar graphs

In this section, we will proof that the thickness of strongly fan-planar graphs is at most three. For bipartite strongly fan-planar graphs, we derive the tight bound of two (which is witnessed for example by a straight-line drawing of $K_{4,n-4}$). In fact, we prove a slightly stronger statement, namely that **any** strongly fan-planar drawing can be colored with at most three colors such that no two edges of the same color cross (respectively, two colors are sufficient for any strongly fan-planar drawing of a bipartite graph). Previously, the best upper bound for the thickness of (bipartite) strongly fan-planar graphs were derived by a combination of their maximum edge-density, Inequality 3.7 and by applying [149] using the fact that (bipartite) strongly fan-planar graphs are hereditary, which gave thickness four and five, respectively.

Our result is obtained by a complete characterization of odd cycles in the intersection graph of a strongly fan-planar drawing, which should be of interest itself.

We begin by describing some specific notation for the remainder of the section.

Definitions We assume that any graph G and its corresponding fan-planar drawing Γ are *simple*. Recall that this implies that G has no self-loops or multiple edges, while in Γ no two adjacent edges cross, any two edges cross at most once and no three edges cross in the same point. We will refer to all of these properties as (S).

Let G be a strongly fan-planar graph with a strongly fan-planar drawing Γ. Since we only consider strong fan-planarity in this section, which implies that none of the patterns shown in Fig. 9.1 occur in Γ, we will simply abbreviate "strongly fan-planar" with "fan-planar" from now to improve the readability. For an edge e_i, we will always denote the anchor of e_i as v_i unless otherwise specified. As stated earlier, we will fully characterize odd cycles in the intersection graph $\mathcal{I}$ of G. In particular, we will consider *chordless* cycles of $\mathcal{I}$, which are cycles without diagonals. Such a chordless cycle C in $\mathcal{I}$ corresponds to a sequence of edges $e_1, \ldots, e_k$, such that e_i and e_{i+1} intersect, but there are no other intersections between the edges of C in G. We will refer to this property as (M). For a fixed chordless cycle C, we will define a local orientation of the edges. Namely, we denote by x_i the intersection point between e_{i-1} and e_i and by a_i and b_i the endpoints of e_i such that $a_i x_i x_{i+1} b_i$ appear in this order on e_i. Vertex a_i is then called the *source* of e_i and b_i the *target* of e_i (with respect to C). Further we denote by $\hat{e}_i$ the oriented segment of e_i from x_i to x_{i+1} which will be called the *base* of e_i. By concatenating the bases $\hat{e}_i$ in order, we obtain a closed loop which will be called $\mathcal{L}$. By (M) we have that $\hat{e}_i$ and $\hat{e}_j$ do not intersect - thus $\mathcal{L}$ forms a closed Jordan Curve which partitions the plane into two regions. Since the edges e_i and e_{i+2} both cross e_{i+1}, they share a common endpoint by (I) which is the anchor v_{i+1} of e_{i+1}. Further, we denote by G_C the subgraph of G induced by the edges of C. An edge e_i of a chordless cycle is called *canonical* if the anchor v_i of e_i is the target of e_{i-1} and the source of e_{i+1}, i.e., $b_{i-1} = a_{i+1}$. If additionally no other edge of C is incident to v_i, i.e., v_i is a degree two vertex in G_C, then e_i is called *strictly canonical*. Fig. 9.12 shows a sequence of canonical edges, however some endpoints of the edges can potentially coincide as shown in Fig. 9.15. For a canonical edge e_i,

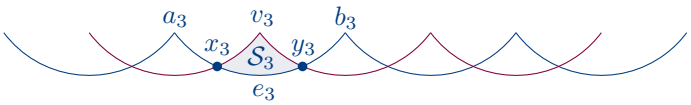


FIGURE 9.12: A sequence of canonical edges.

we will call the triangular region $\mathcal{S}_i$ with corners x_i, x_{i+1}, v_i and delimited by the

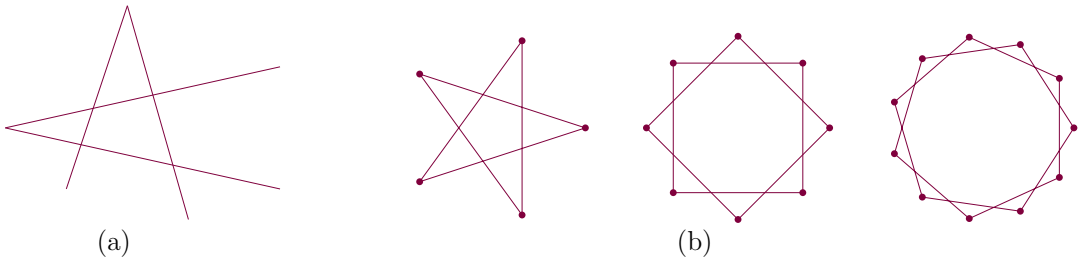


FIGURE 9.13: (a) The only possible cycle of length four, and (b) fully canonical cycles for $k = 5, 8, 11$.

edges e_{i-1}, e_i, and e_{i+1} the *spike* of e_i, refer to the shaded region in Fig. 9.12. If a chordless cycle C only contains canonical edges, then C is called *fully canonical*.

Such cycles can be created for example for any $k \geq 5$ by taking the corners of a regular k-gon and connecting every other corner with an edge, see Fig. 9.13(b). Note that such a fully canonical cycle in $\mathcal{I}$ corresponds to a single closed trail of length k in G for odd k, but to two closed trails of length $k/2$ for even k. These closed trails in$G$ are not necessarily cycles, as the anchors of the edges of a fully canonical cycle can coincide as discussed earlier, see e.g. Fig. 9.14(left).

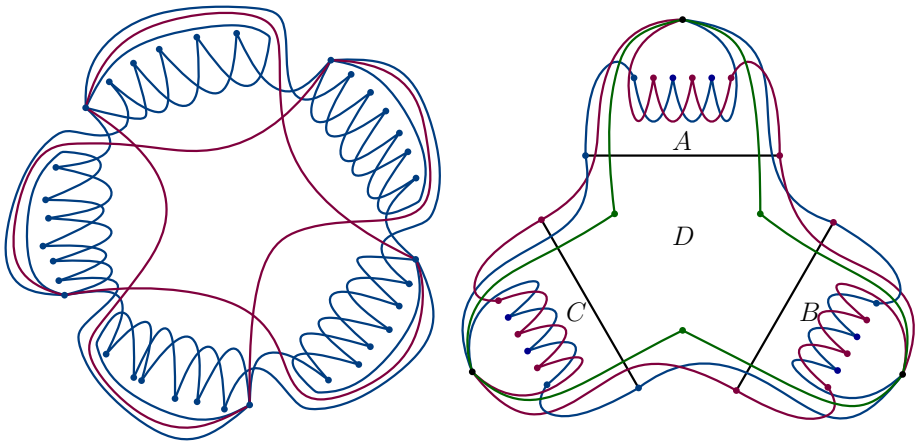


FIGURE 9.14: Chordless cycles can intersect and share edges.

9.3.1 Characterizing chordless cycles

In the following, we provide a full characterization of chordless cycles which is a key ingredient for the main result of this section. Before going into the technical details of the section, let us first consider some examples how chordless cycles of the intersection graph can interact with each other. Fig. 9.14 shows two examples of fan-planar graphs whose intersection graphs have several chordless cycles that cross and share edges. The graph on the left has $2^5 = 32$ distinct chordless cycles, as for each of the five red edges we can instead traverse the corresponding blue edges. In the graph on the right, the boundaries of the faces labeled A, B, C are loops of chordless cycles of length 11. There is a chordless cycle of length 9 surrounding face D, there is a chordless cycle of length 30 surrounding all four faces, and there are three chordless cycles of length 23 surrounding ABD, BCD, and CAD, respectively.

We first observe that by (I), no two independent edges can cross a common edge. Moreover, by (S), no two adjacent edges cross and therefore $\mathcal{I}$ does not contain cycles of length three. Cycles of length four have a unique shape, see Fig. 9.13(a).

Hence, it remains to study chordless cycles of length at least five. The following lemma guarantees that the anchors of a small number of consecutive canonical edges are different.

Lemma 26. *Let $e_1, \dots, e_5$ be five consecutive canonical edges of a chordless cycle. Then the anchors of the five edges are distinct.*

Proof. Recall that v_i is the anchor of e_i. Since v_2 is incident to e_1, it follows by (S) that $v_2 \neq v_1$. Since $e_1, \dots, e_5$ are canonical, it follows that v_1 and v_3 are the endpoints of e_2 and since G contains no self-loop we have that $v_1 \neq v_3$. If $v_1 = v_4$ would hold, then e_2 and e_3 would share an endpoint (as $v_1 = a_2$ and $v_4 = b_3$) but also cross by assumption, a contradiction to (S). Finally, if $v_1 = v_5$, then $e_2 = (v_1, v_3)$ and $e_4 = (v_3, v_5)$ would be parallel edges, a contradiction to (S). Using analogous arguments, we can derive the remaining inequalities which concludes the proof. □

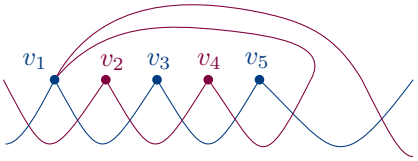


FIGURE 9.15: The anchors of canonical edges can coincide, but only after at least four distinct anchors.

The following lemma ensures that an anchor v_i cannot be the source/target for both e_{i-1} and e_{i+1} simultaneously.

Lemma 27. *Let $C = (e_1, \dots e_k)$ be a chordless cycle in $\mathcal{I}$ with $k \geq 5$. Then, $b_{i-1} \neq b_{i+1}$ and $a_{i-1} \neq a_{i+1}$ for $1 \leq i \leq k$.*

Proof. Suppose for a contradiction that there exists an index i such that $b_{i-1} = b_{i+1}$. By applying a cyclic renumbering of the indices, we can assume that $b_1 = v_2 = b_3$. Due to the orientation of the edges with respect to C, we have that x_4 lies on the segment $x_3b_3 = x_3v_2$, while x_1 lies on the segment a_1x_2, refer to Fig. 9.16 for an illustration of the steps throughout the proof. The edge e_4 which contains x_4 is by definition incident to v_3. If $v_3 = a_2$ would hold, then either e_4 would cross e_1, which is not allowed by (M), or e_3 would form (II) together with e_2 and e_4 - hence $v_3 = b_2$ holds. Consider the region $\mathcal{R}$ bounded by $v_2x_2x_3$. The (open) segments v_2x_2 and x_2x_3 do not cross any edge of C by (M) and segment x_3v_2 only crosses e_4 in x_4. Hence, the other endpoint of e_4 different from v_3 lies inside $\mathcal{R}$. The intersection point x_5 cannot lie inside $\mathcal{R}$, as otherwise the closed curve $\mathcal{L}$, which has to return to x_1, would necessarily intersect $\mathcal{R}$ again, which is impossible as established earlier. Thus, x_5 which lies on the segment x_4b_4 of e_4 lies outside $\mathcal{R}$ and hence $b_2 = v_3 = b_4$. Repeating this same argument would imply $b_5 = b_3 = b_1$, $b_6 = b_4 = b_2$, and so on. Consider now the final edge e_k of C. By construction, e_k contains x_1 and therefore it cannot be incident to b_1 by (S). Thus, it is incident to b_2 and x_k lies on the edge e_{k-1} incident to b_1 since they are required to alternate. Since x_1 lies on e_k on the segment x_kb_k with $b_k = b_2$, we get two possibilities for drawing e_k, the matching part of both curves is drawn solid, while the diverging parts are drawn dashed and dotted in Fig. 9.16. The dotted version violates (III) for e_{k-1} (with e_{k-2} and e_k) while the dashed version violates (III) for e_k (with e_1 and e_{k-1}) and we obtain a contradiction.

We now assume that $a_{i-1} = a_{i+1}$ holds for some i for our chordless cycle C. We denote by $C' = (e_k, \dots, e_1)$ the chordless cycle which is obtained by reversing C. In

particular, this reversing flips the source and target for each edge, so C' now has an index j (which is in fact $k - i$) where $b_{j-1} = b_{j+1}$ which we already showed cannot exist.

□

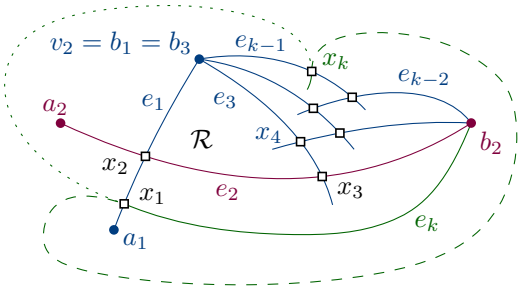


FIGURE 9.16: The cycle cannot be closed.

Corollary 11. *Let e_i be a non-canonical edge of a chordless cycle C. Then $a_{i-1} = v_i = b_{i+1}$.*

Proof. Anchor v_i is by (I) a common endpoint of e_{i-1} and e_{i+1}. Since e_i is not canonical by assumption, we have that $b_{i-1} \neq a_{i+1}$. By Lemma 27 we have $b_{i-1} \neq b_{i+1}$ and $a_{i-1} \neq a_{i+1}$, which then implies $v_i = a_{i-1} = b_{i+1}$. □

The last technical lemma of this subsection verifies the existence of a sequence of consecutive canonical edges in every sufficiently large chordless cycle.

Lemma 28. *A chordless cycle of length at least five has at least four consecutive canonical edges.*

Proof. Let $C = (e_1, \dots e_k)$ be a chordless cycle of length $k \geq 5$ such that $e_1, \dots, e_m$ is a longest consecutive sequence of canonical edges in C, and assume for a contradiction that $m < 4$. Based on the value of m, we distinguish the following cases, which are visualized in Fig. 9.17.

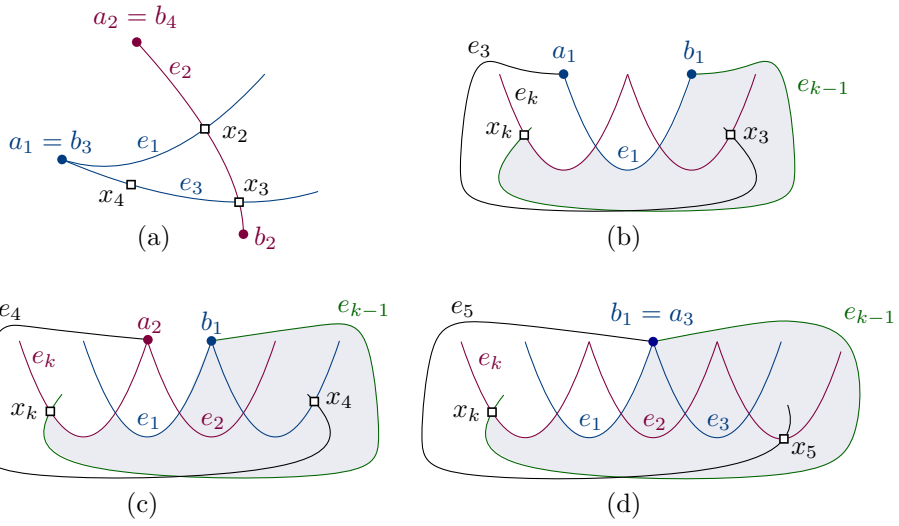


FIGURE 9.17: (a) $m = 0$, (b) $m = 1$, (c) $m = 2$ and (d) $m = 3$.

- $m = 0$

  This implies that C does not contain any canonical edge. By Corollary 11 it follows that $a_1 = v_2 = b_3$ and $a_2 = v_3 = b_4$. By definition, we have that x_4 lies on the segment x_3b_3 and it is contained in e_4 whose endpoint is $b_4 = a_2$. However, e_4 cannot intersect e_1 by (M), and it cannot intersect e_2 by (S). But then it necessarily forms (II) with e_2 and e_3, see Fig. 9.17(a).

- $m = 1$

  Assume w.l.o.g. that e_1 is canonical. Then, e_2 and e_k are not canonical by assumption. Corollary 11 then establishes that $a_1 = b_3$ and $a_{k-1} = b_1$. Since e_3 does not cross e_k and e_2 does not cross e_{k-1} by (M) and since x_k is contained in e_{k-1} and $x_3 \in e_3$, we have exactly the situation shown in Fig. 9.17(b) by (II) and (III). But then e_3 and e_{k-1} necessarily cross and thus $k = 5$ has to hold to not obtain a contradiction to (M). As both e_3 and e_5 cross e_4 by construction, they have a common endpoint. This common endpoint cannot be a_1, as clearly e_1, which is crossed by e_5, is also incident to a_1 which violates (S). The other endpoint of e_5 is contained in a region delimited by $e_3e_{k-1}e_ke_1$ - if e_3 would try to enter this region it would either cross itself, cross e_{k-1} twice, cross e_k or cross e_1 which is also incident to a_1, hence in any case we violate (S) or (M).

- $m = 2$

  W.l.o.g. assume that e_1 and e_2 are canonical. Since e_3 and e_k are then not canonical, by Corollary 11 we have $a_2 = b_4$ and $b_1 = a_{k-1}$. Similar to the previous case, e_{k-1} and e_3 do not cross and since $x_k \in e_{k-1}$ and $x_4 \in e_4$ we have the situation of Fig. 9.17(c). In particular, e_4 and e_{k-1} cross which implies $k = 6$. But then e_4, e_5 and e_6 form (II), a contradiction.

- $m = 3$

  Again, assume w.l.o.g. that e_1, e_2 and e_3 are canonical, which implies by assumption that e_4 and e_k are not canonical. By Corollary 11 we have that $a_3 = b_5$ and $b_1 = a_{k-1}$. The edge e_2 is canonical and thus $a_{k-1} = b_1 = a_3 = b_5$ which implies that e_5 and e_{k-1} cannot cross, hence $k \neq 5$. If $k = 6$, then $a_5 = b_5$, which is impossible, hence $k \geq 7$. By (M) we have that e_5 does not cross $\{e_1, e_2, e_3, e_k\}$ while e_{k-1} does not cross $\{e_1, e_2, e_3, e_4\}$. Therefore $x_k \in e_{k-1}$ and $x_5 \in e_5$ implies that we have the situation of Fig. 9.17(d). But here we have that e_5 and e_{k-1} cross, a contradiction which concludes the proof.

□

With these tools at hand, we are ready to state the main result of this subsection which is the characterization of chordless cycles that are not fully canonical.

Theorem 26. *If a chordless cycle of length $k \geq 5$ is not fully canonical, then $k \geq 9$, edges $e_1, \ldots, e_{k-1}$ are canonical, anchors $v_2 = v_{k-2}$ coincide so that $b_1 = a_3 = b_{k-3} = a_{k-1}$, and b_{k-1} and a_1 are vertices of degree one in G_C.*

Proof. Let $C = (e_1, \ldots, e_k)$ be a chordless cycle of length $k \geq 5$ which is not fully canonical and let $e_1, \ldots, e_{m-1}$ be the longest sub-sequence of C which consists of canonical edges. Lemma 28 establishes that $m \geq 5$. Since e_m is not canonical, Corollary 11 implies that $a_{m-1} = b_{m+1}$. Recall that by definition, e_{m+1} crosses e_m in x_{m+1} and no other edge besides e_{m+2} by (M). Denote by $\mathcal{R}$ the region enclosed by $\hat{e}_m$ and the edges e_{m-1} and e_{m+1}. As $\mathcal{R}$ cannot contain an endpoint of e_m since (II) and (III)

are not allowed, it follows that we have the situation shown in Fig. 9.18(a), where the region $\mathcal{R}$ is shaded in yellow.

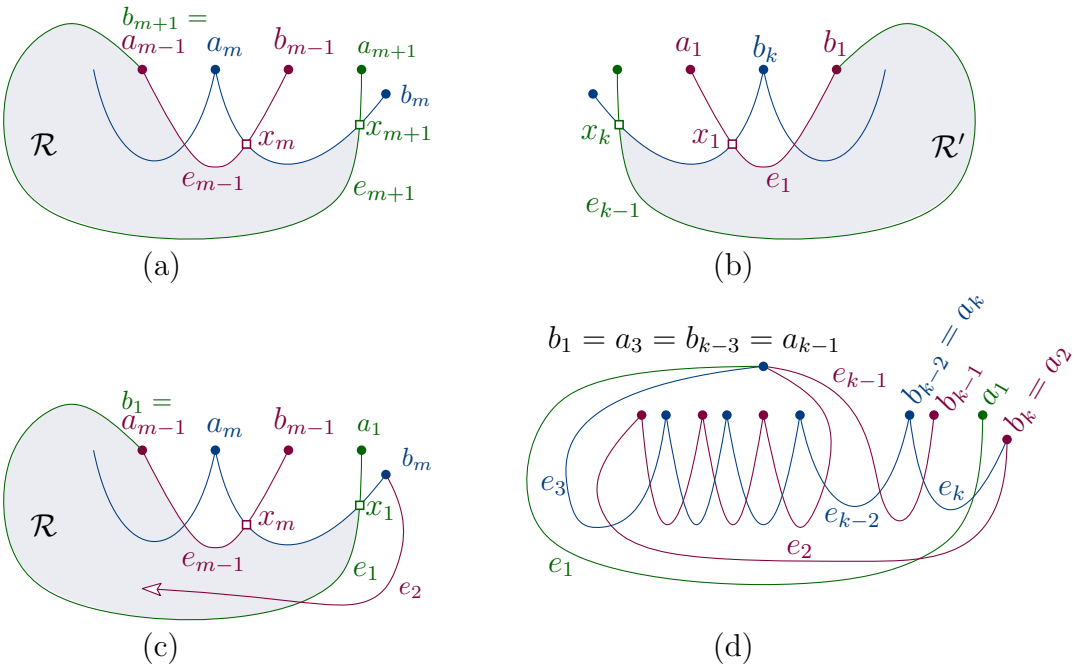


FIGURE 9.18: Proof of Theorem 26.

By (M), it follows that the closed curve $\mathcal{L}$ of C lies entirely in $\mathcal{R}$ with the exception of the bases $\hat{e}m-1, \hat{e}m$ and $\hat{e}m+1$ which lie on its boundary. We want to establish that $k = m$. Clearly, $k \geq m$ has to hold, thus assume for a contradiction that $k > m$ If $k = m+1$, we have $e_k = e_{m+1}$. Since e_k is not canonical (as otherwise, the subsequence $e_k, e_1, \ldots, e_{m-1}$) would be the maximal one), it follows that $a_{k-1} = a_m = b_1$ and e_1 intersects e_k between x_{m+1} and $b_{m+1} = a_{m-1}$. As e_1 cannot intersect e_{m-1} by (M), the crossing between e_1 and e_k would violate either (II) or (III).

If $k = m+2$, then $e_{k-1} = e_{m+1}$ in Fig. 9.18(a), x_k lies on e_{m+1} on the boundary of $\mathcal{R}$, and $\hat{e}_k$ lies (except for its endpoint x_k) in the interior of $\mathcal{R}$ (in particular, x_1). Again, since e_k is not canonical, we have $a_{k+1} = b_1$ by Corollary 11 and since e_1 does not cross the boundary of $\mathcal{R}$, it cannot contain x_1 which lies in the interior of $\mathcal{R}$

Assume now that $k > m+2$. Then, $\hat{e}_k$ is completely contained in the interior of $\mathcal{R}$ Since e_k is not canonical, we have $a_{k-1} = b_1$. Symmetrically to the previous argument, the edge e_{k-1} must be such that the region $\mathcal{R}'$ formed by $\hat{e}_k$, e_{k-1}, and e_1 contains no endpoint of e_k to avoid (II) and (III), so we are in the situation of Fig. 9.18(b). Again, the loop $\mathcal{L}$ lies in $\mathcal{R}'$, with only $\hat{e}_{k-1}, \hat{e}_k, \hat{e}_1$ on the boundary of $\mathcal{R}'$. In particular, $\hat{e}_m$ lies in the interior of $\mathcal{R}'$. Thus we have the following situation. Base $\hat{e}_m$ lies in the interior of $\mathcal{R}'$ but on the boundary of $\mathcal{R}$ and base $\hat{e}_k$ lies in the interior of $\mathcal{R}$ but on the boundary of $\mathcal{R}'$. This implies that the boundaries of $\mathcal{R}$ and $\mathcal{R}'$ must intersect. The boundary of $\mathcal{R}$ consists of e_{m-1}, $\hat{e}_m$, and e_{m+1}, the boundary of $\mathcal{R}'$ consists of e_{k-1}, $\hat{e}_k$, and e_1. By (M), the only allowed edge crossing occurs when $k = m+3$ such that e_{m+1} and $e_{m+2} = e_{k-1}$ cross - however since the two closed curves $\mathcal{R}$ and $\mathcal{R}'$ need to cross an even number of times (hence, at least twice) and since any pair of edges crosses at most once by (S), it follows that $\mathcal{R}$ and $\mathcal{R}'$ require a common vertex for the other "intersection" and thus $a_{m-1} = b_1$. But then, e_{m-1} and e_{k-1} share a common vertex and thus cannot cross at all by (S).

It follows that our assumption that $k > m$ is false, and so $k = m$ holds. After an appropriate relabeling of the indices modulo k, we obtain the situation of Fig. 9.18(c). Recall that so far we established that C has exactly one edge e_m which is not canonical. Since e_1 is canonical by assumption we have $b_m = a_2$. Thus, the source of e_2 is b_m and it enters region $\mathcal{R}$ through e_1. By (M), e_2 crosses neither e_{m-1} nor e_m, so its other endpoint b_2 necessarily lies in the interior of $\mathcal{R}$. As the sub-sequence of $e_2, \dots, e_{k-1}$ is canonical, we end up with the situation shown in Fig. 9.18(d).

Since $v_2 = b_1 = a_{k-1} = v_{k-2}$, Lemma 26 implies that $k - 2 \geq 7$, so $k \geq 9$. □ □

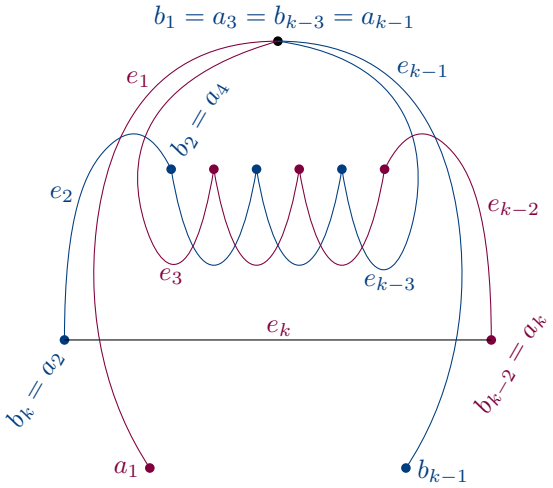


FIGURE 9.19: A non-canonical chordless cycle.

Fig. 9.19 is an alternative drawing of the non-canonical chordless cycle of Fig. 9.18(d), showing the symmetry in the characterization. We also remark the following observation, which will be important for the proof of Corollary 12.

Observation 5. *Any odd cycle in $\mathcal{I}$ implies an odd-length cycle in G*

Proof. The statement is obvious if the odd cycle is fully canonical. For a non-canonical cycle $C = (e_1, \dots, e_k)$, we can observe by the characterization derived in Theorem26 that there is a cycle of length $k - 2$ which consists of all edges of C except for e_{k-1} and e_1. In particular, the cycle is $e_2, e_4, e_6, \dots, e_{k-3}, e_3, e_5, \dots, e_{k-2}$. Since k is by assumption odd, the result follows. □

We can now derive the first major result of this section.

Corollary 12. *The edges of a strongly fan-planar drawing of a bipartite graph $\mathcal{G}$ can be colored using two colors such that no edges of the same color cross. As a consequence, a bipartite, strongly fan-planar graph has thickness at most two.*

Proof. We show that $\mathcal{I}$ is bipartite. Assume otherwise: then $\mathcal{I}$ has an odd cycle, which contains a chordless odd cycle C of length k. By Observation 5, this implies an odd cycle in G, hence G is not bipartite and we obtain contradiction. □

We remark here the the bound on the thickness is tight as witnessed for example by the non-planar graph $K_{3,3}$ which admits a drawing with one crossing and it is therefore also strongly fan-planar.

For the remainder of this subsection, with the full characterization at hand, we will study additional properties of chordless cycles. In particular, we are concerned with the interaction of the region induced by the closed curve $\mathcal{L}$ and the spikes $\mathcal{S}_i$

with *other* edges which are not part of C. Recall that the closed curve $\mathcal{L}$ partitions the plane into two regions. If C is fully canonical, then one of these regions is empty in G_C—this could be either the bounded or the unbounded region of the plane delimited by $\mathcal{L}$. Otherwise, C has exactly one non-canonical edge by Theorem 26 and the *bounded* region delimited by $\mathcal{L}$ is empty in G_C, i.e., it contains no vertices of G_C. With a slight abuse of notation, we will in both cases denote the empty region that is delimited by $\mathcal{L}$ by $\mathcal{L}$ as well. By construction, for each base $\hat{e}_i$ of a canonical edge e_i, we have that the (empty) region defined by spike $\mathcal{S}_i$ is adjacent to $\mathcal{L}$.

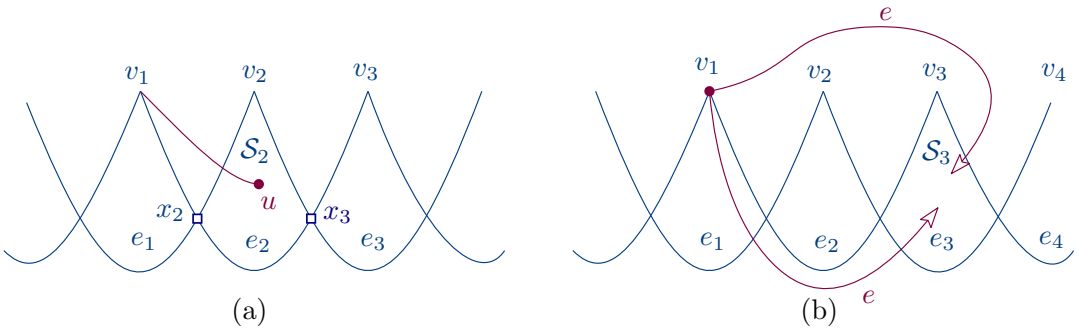


FIGURE 9.20: (a) u must lie in $\mathcal{S}_2$ or $e = e_2$, and (b) e starting in v_1 cannot enter $\mathcal{S}_3$.

The following three lemmas consider the interaction between spikes and additional edges.

Lemma 29. *Let e_1, e_2, e_3 be three consecutive canonical edges of a chordless cycle C. Let $e = (v_1, u)$ be an edge that crosses e_1 in the relative interior of the segment x_2v_2, i.e., $e \notin C$. Then u is contained in $\mathcal{S}_2$.*

Proof. Traversing e from v_1 to u, e enters $\mathcal{S}_2$ by crossing the segment x_2v_2 by assumption see Fig. 9.20(a). Edge e cannot cross e_2 or cross e_1 again by (S), hence it could only leave $\mathcal{S}_2$ through the segment v_2x_3. But then e has to be incident to v_3, which by Lemma 26 is different from v_1. Thus, $u = v_3$ which implies by (S) that $e = e_2$, a contradiction as e_2 does not intersect the relative interior of x_2v_2. □

Lemma 30. *Let e_1, e_2, e_3, e_4 be four consecutive canonical edges of a chordless cycle. Let $e = (v_1, u)$ be an edge incident to v_1. Then e does not enter the interior of $\mathcal{S}_3$.*

Proof. Recall that by definition, $\mathcal{S}_3$ is delimited by v_2x_3, $\hat{e}_3$ and v_3x_4, see Fig. 9.20(b). By (S) e, which is incident to $v_1 = a_2$, cannot cross e_2, hence in order to enter the spike it has to intersect either $\hat{e}_3$ or v_3x_4. In order to cross $\hat{e}_3$, e has to be incident to v_3. Since $v_1 \neq v_3$ by Lemma 26, it follows that $e = (v_1, v_3) = e_2$ by (S), which does not intersect the interior of $\mathcal{S}_3$ by definition. Finally, if e intersects e_4, we have that e is incident to v_4. Since $v_1 \neq v_4$ holds by Lemma 26, we have $e = (v_1, v_4)$. By construction, vertex v_4 lies outside $\mathcal{S}_3$, hence e would have to cross e_4 again which is a contradiction to (S). □

Lemma 31. *Let C be a chordless cycle of length at least five, and let e be an edge not part of C such that e intersects the loop $\mathcal{L}$ of C. Then e starts in the anchor v_i of a canonical edge e_i of C, passes through the spike $\mathcal{S}_i$, crosses the base $\hat{e}_i$, and either (1) ends in $\mathcal{L}$; or (2) crosses the base $\hat{e}_j$ of another canonical edge e_j of C, then passes through $\mathcal{S}_j$ and terminates in v_j; or (3) crosses the base of the non-canonical edge e_j of C, never enters $\mathcal{L}$ again, and terminates in a vertex that is not a vertex of C. The second and third case can only happen if e_i and e_j share*

an endpoint. In particular, (2) cannot happen when e_{i-1} and e_{i+1} are strictly canonical, and (3) implies that $i \in \{j-2, j+2\}$.

Proof. We begin with a preliminary observation. If an edge e intersects the closed curve $\mathcal{L}$ in a point p on a canonical edge e_i, such that $p \in \hat{e}_i$, then we have by (I) that the anchor v_i of e_i must be an endpoint of e. Further, we have by (S) and (II) that the part of e between v_i and p lies completely in the spike $\mathcal{S}_i$.

We first consider the case where e does not cross a non-canonical edge. If we assume that e crosses $\mathcal{L}$ exactly one in a canonical edge e_i, then by the observation above we necessarily have the following situation: e starts in v_i, passes through $\mathcal{S}_i$, crosses $\hat{e}_i$ and its other endpoint lies inside $\mathcal{L}$. If e crosses the curve $\mathcal{L}$ more than once, say it crosses edge e_j in addition to e_i,then by assumption it follows that e_j is also canonical. But then we can repeat the initial observation and obtain that e starts in v_i, passes through $\mathcal{S}_i$, crosses $\hat{e}_i$, passes through $\mathcal{L}$, crosses base $\hat{e}_j$, passes through $\mathcal{S}_j$, and ends in v_j. Since e intersects both e_i and e_j, we have by (I) that they share a common endpoint.

If e crosses $\mathcal{L}$ more than once, then by the above this must be in two canonical edges e_i and e_j. Then we are in case (2): e starts in v_i, passes through $\mathcal{S}_i$, crosses $\hat{e}_i$, passes through $\mathcal{L}$, crosses base $\hat{e}_j$, passes through $\mathcal{S}_j$, and ends in v_j. Since e intersects e_i and e_j, (I) implies that e_i and e_j have a common endpoint. If e_{i-1} and e_{i+1} are strictly canonical, then we have that the endpoints of e_i are the anchors of e_{i-1} and e_{i+1} and both these vertices have degree two in G_C; thus we would have that $j \in \{i-2, i+2\}$. But then e is identical to either e_{i-1} or e_{i+1}, a contradiction which ensures the latter part of statement (2).

Now we consider the case where e crosses the non-canonical edge e_j of C in a point q. By (I) we have that the anchor v_j is an endpoint of e. Moreover, (II) implies that the part of e connecting v_j and q has to lie inside the region delimited by $ej-1, \hat{e}_j$ and e_{j+1}, refer to Fig. 9.19 with $j = k$. But then this part passes through $\mathcal{L}$, hence it has to cross another edge e_i of C which is canonical. By using the initial observation once again, we have the situation that e starts in v_i, passes through $\mathcal{S}_i$, crosses $\hat{e}_i$, passes through $\mathcal{L}$, and crosses $\hat{e}_j$. Clearly, e cannot cross $\hat{e}_j$ again by (S) and it cannot cross another canonical edge, hence e does not enter $\mathcal{L}$ again. In particular, this implies that $v_i = v_j$ has to hold. Similar to the previous case, e_i and e_j, which are both crossed by e, need to have a common endpoint by (I). By Theorem 26, the only edges that share an endpoint with the non-canonical edge e_j are e_{j-2} and e_{j+2} which establishes $i \in \{j-2, j+2\}$.

Finally, we will show that the other endpoint of e different from $v_i = v_j$ is not part of C. Once e leaves $\mathcal{L}$ by crossing $\hat{e}_j$, it can only reach (i) the endpoints of e_j, which is a contradiction to (S), (ii) the anchor v_j of e_j but then e_j would be a self-loop or (iii) the endpoints b_{j-1} and a_{j+1}, again, refer to Fig. 9.19 with $j = k$. In the last case, we observe that since $v_i = v_j = a_{j-1} = b_{j+1}$ as e_j is not canonical, it follows that $e = (a_{j-1}, b_{j-1})$ or $e = (b_{j+1}, a_{j+1})$, but then e is identical to either e_{j-1} or e_{j+1}, a contradiction. □

9.3.2 Coloring with Three Colors

With most of the technical tools at hand, we can now state the main theorem of this section, which directly implies the desired result. To enable an easy cross-referencing, we will reuse "strongly fan-planar" for the following statements.

Theorem 27. *In every strongly fan-planar drawing of a graph $\mathcal{G}$ there is a set S of edges such that (1) S is independent in $\mathcal{I}$, that is, no two edges in S cross; and (2) every odd cycle in $\mathcal{I}$ contains an edge in S.*

Corollary 13. *The edges of a strongly fan-planar drawing of a graph $\mathcal{G}$ can be colored using three colors such that no two edges of the same color cross. As a consequence, a strongly fan-planar graph has thickness at most three.*

Proof. Pick the set of edges S according to Theorem 27 and color the edges of S using one color. Then, $\mathcal{I} \setminus S$ contains no odd cycles and hence admits a proper coloring using the two remaining colors. □

In the following, we will describe the construction of such a set S for the proof of Theorem 27 which makes use of the following lemma.

Lemma 32. *Let C be a chordless cycle of length at least five. Then C contains an edge e_i such that*

- *$e_{i-2}, e_{i-1}, e_i, e_{i+1}, e_{i+2}$ are all canonical in C and*
- *e_{i-1}, e_i, e_{i+1} are all strictly canonical in C.*

Proof. If C is fully canonical, any edge satisfies the aforementioned criteria as every edge is canonical and by Lemma 26 we have that the anchors of a canonical sequence can only coincide if they are sufficiently far apart. If C is not canonical, by Theorem 26 we have that the sub-sequence $e_2, \dots, e_{k-2}$ fulfills the following properties. On the one hand, we have $v_2 = v_{k-2}$. On the other hand, the spikes $\mathcal{S}_3, \mathcal{S}_4, \dots, \mathcal{S}_{k-3}$ are completely contained in the region delimited by $e_1, \hat{e}_k$ and e_{k-1}. Since spikes cannot intersect (they can only potentially touch at the anchors) by (I), we have that the corresponding anchors form a nested structure - pick the innermost interval $e_j, \dots, e_\ell$ such that $v_j = v_\ell$. Since this is the innermost interval, the anchors $v_j, \dots, v_{\ell-1}$ are not involved in a nesting and therefore distinct. By Lemma 26 we necessarily have that $\ell - j \geq 5$. Thus by choosing $i = j + 2$ we can satisfy the requirements of the lemma. □

For any chorldess cycle C, we will pick an edge e of C which satisfies the properties of Lemma 32. Edge e will be called the *ground edge* of C. The set s of Theorem 27 will contain exactly a subset of all these ground edges. In particular, the following key lemma establishes that if two such ground edges intersect (i.e., they would be adjacent in $\mathcal{I}$) then they actually belong to the other chordless cycle, hence it is sufficient to simply select one of them to break both cycles. As usual, the proof requires some technical arguments and relies on the following two lemmas which we have to establish before.

Lemma 33. *Let C and C' be two chordless odd cycles with ground edges e and e' and such that e and e' cross in a point z. Then z does not lie in the relative interior of the base $\hat{e}$.*

Proof. Denote the edges of C by $e_1 \dots e_k$ and the edges of C' by $e'_1, \dots, e'_\ell$. After a cyclic renumbering, we can assume that the ground edges e and e' coincide with e_3 and e'_3, respectively.

Assume for a contradiction that point z lies in the interior of $\hat{e}_3$. Since e_3 is strictly canonical, it follows by Lemma 31 that e'_3 starts at v_3, passes through $\mathcal{S}_3$, crosses $\hat{e}_3$ and its other endpoint lies inside $\mathcal{L}$ without crossing it again (note that (2) and (3) cannot occur as e_3 is a ground edge). On the other hand, since e_3 crosses e'_3, one endpoint of e_3 is v'_3. We assume that $b_3 = v_4 = v'_3$, the other case is symmetric. We can further assume w.l.o.g that v_3 is the source a'_3 of e'_3, the other case can be handled by simply reversing the orientation of C'. Consider the edge e'_4, which starts at $v'_3 = v_4$. By definition, e'_3 and e'_4 intersect in x'_4 that lies on the edge e'_3, hence either inside $\mathcal{L}$ or inside $\mathcal{S}_3$. For the former case, we observe that e'_4 must intersect e_4 to enter $\mathcal{L}$ by Lemma 31, as it is incident to $b_3 = v_4$. For the latter case, we can observe that e'_4 cannot enter $\mathcal{S}_3$ by crossing e_2, as otherwise e_2, e_3 and e'_4 would form (II). Hence, it also to cross e_4 in order to enter. But then the crossing of e_4 and e'_4 implies that v'_4 is an endpoint of e_4. Recall that we assumed that $v_3 = a'_3$ and thus it follows that $v_3 = v'_2$. Since $v'_2 \neq v'_4$ by Lemma 26, it follows that $e_4 = (v'_2, v'_4) = e'_3$. But then $z = e_3 \cap e'_3 = e_3 \cap e_4 = x_4$ which means that z does not lie in the relative interior of $\hat{e}$, a contradiction. □

The following lemma now establishes that such an intersection point z in fact has to lie in base $\hat{e}$ which then together with Lemma 33 implies that e' coincides with an edge of C.

Lemma 34. *Let C and C' be two chordless odd cycles with ground edges e and e' and such that e and e' cross in a point z. Then z lies in base $\hat{e}$.*

To make the following proof more readable, we again proof a rather technical lemma, whose setting is visualized in Fig. 9.21, beforehand.

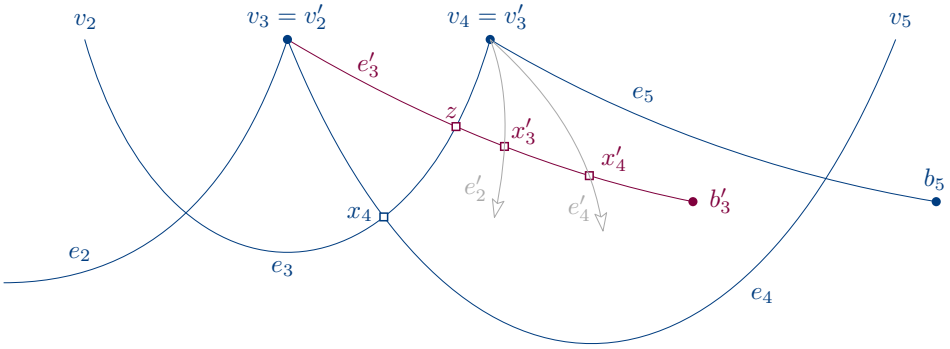


FIGURE 9.21: Proof of Lemma 35.

Lemma 35. *Let C and C' be two chordless odd cycles with ground edges e_3 and e'_3. Suppose that $v_3 = v'_2$, $v_4 = v'_3$, that e'_3 intersects x_4v_4 such that $b'_3 \in \mathcal{S}_4$, and that $x'_3 \in \mathcal{S}_4$. Then the entire loop $\mathcal{L}'$ of C' lies in $\mathcal{S}_4$.*

Proof. Assume for a contradiction that $\mathcal{L}'$ contains a point outside of $\mathcal{S}_4$. Follow the closed curve of $\mathcal{L}'$ starting at x'_4, which lies (strictly) inside $\mathcal{S}_4$ as it is contained in the segment zb'_3 where b'_3 is inside $\mathcal{S}_4$ by assumption and since z, by definition, lies on the boundary of $\mathcal{S}_4$. Denote by e'_m the first edge of C' such that x'_m is still inside $\mathcal{S}_4$, but $\hat{e}'_m$ intersects the boundary of $\mathcal{S}_4$. Since $\mathcal{S}_4$ is bounded by e_3, e_4 and e_5, the edge e'_m has to intersect one of them, see Fig. 9.21.

- $\hat{e}'_m$ intersects e_3 in point p

  Then e'_m is incident to $v_3 = v'_2$. By the choice of the ground edge e'_3, we have that e'_2 is strictly canonical, hence v'_2 has degree two in $G_{C'}$. Hence, $e'_m = e'_1$

follows. Edge e'_1 cannot intersect e_4 by simplicity, as they are both incident to $v_3 = v'_2$. Observe that p cannot lie on e_3 between z and $v_4 = v'_3$, as otherwise e'_1 would necessarily intersect e'_4, a violation of (M). It follows that point p lies on e_3 between x_4 and $z = e_3 \cup e'_3$. Further, we have that x'_2 lies in the triangle formed by e_3, e_4 and e'_3 by (S). If x'_2 and p coincide, we have $e'_2 = e_3$ which implies that $x'_3 = z$ and then $\mathcal{L}'$ does not leave $\mathcal{S}_4$. Otherwise, if x'_2 lies strictly outside $\mathcal{S}_4$, then $e'_2 \neq e_3$ holds and thus x'_3 lies strictly inside $\mathcal{S}_4$ by assumption. But then the edge e'_2 intersects the boundary of $\mathcal{S}_4$ – it cannot cross e_3 or e_5 by (S) and since it contains x'_3 in the interior of $\mathcal{S}_4$ (while $x'_2 \in e'_2$ lies outside $\mathcal{S}_4$), it would need to intersect e_4 twice, which is impossible.

- $\hat{e}'_m$ crosses e_4

  It follows that e'_m is incident to $v_4 = v'_3$. Again by the choice of our ground edge, v'_3 has degree two in $G_{C'}$ which implies that $e'_m = e_4$. Hence, e_4 crosses e'_4 and is therefore incident to v'_4. As $v_3 = v'_2 \neq v'_4$ holds, it follows that $e_4 = (v'_2, v'_4) = e'_3$. In particular, this implies that b'_3 is then not contained inside $\mathcal{S}_4$, a contradiction.

- $\hat{e}'_m$ crosses e_5

  Again, we have that e_5 is incident to v'_m, i.e., $v'_m = a_5 = v'_3$ or $v'_m = b_5$. For the former case, since e'_3 is strictly canonical, vertex v'_3 is the anchor of exactly one edge, and thus $v'_m = v'_3$ would follow, a contradiction. Hence, we have $v'_m = b_5$. Observe that b_5 lies outside $\mathcal{S}_4$. Since $\hat{e}'_m$ is the first segment of $\mathcal{L}'$ that intersects the boundary of $\mathcal{S}_4$, we have that x'_m is still contained inside $\mathcal{S}_4$. As e'_{m-1} contains x'_m and is incident to b_5 by (I), it follows that e'_{m-1} has to intersect the boundary of $\mathcal{S}_4$. By (S) it cannot intersect e_5. If e'_{m-1} crosses e_4, then e'_{m-1} is also incident to $v_4 = a_5$ and hence is equivalent to e_5. By construction, e_5 contains no point inside $\mathcal{S}_4$, a contradiction. Hence, assume that e'_{m-1} crosses e_3. By (M), it cannot cross e'_2 or e'_3, so the only way to enter $\mathcal{S}_4$ is to cross e_3 on the segment zv_4. However, then the edge has to end inside the triangular region defined by e_3, e'_3 and e'_2. Since $\mathcal{L}$ does not enter this region, as it contains no point on $a'_3x'_3 = v_3x'_3$ or $X'_3b'_2 = x'_3v_4$, we obtain a final contradiction that concludes the proof.

□

We are now ready for the main part of the proof.

Proof of Lemma 34. We will reuse the notation introduced in Lemma 33 and assume for a contradiction that $z \notin \hat{e}$. We further assume that $z \in x_4b_3$, the other case is symmetric. Since e'_3 crosses e_3, e'_3 has an endpoint in v_3. By Lemma 29, this means that the other endpoint u of e'_3 lies in $\mathcal{S}_4$, see Fig. 9.22(a). As e_3 crosses e'_3 we have that v'_3 is an endpoint of $e_3 = (v_2, v_4)$. Assume first that $v'_3 = v_2$. Consider the edges e'_2 and e'_4, which are both incident to $v'_3 = v_2$ by construction. Lemma 30 establishes that they cannot intersect the interior of $\mathcal{S}_4$, therefore x'_3 and x'_4 both lie on the segment v_3z. In order to not violate (II) or (III), they both need to cross e_4 which implies that both are incident to v_4. But then $e'_2 = e'_4$, a contradiction. Thus, we have $v'_3 = v_4$. After a possible reorientation of C', we can w.l.o.g. assume that v_3 is the source a'_3 of e'_3. We now consider the order of x'_3, x'_4, and z on e'_3. By Lemma 33, either z occurs after x'_3 and x'_4 (possibly with $z = x'_4$), or before x'_3 and x'_4 (possibly with $z = x'_3$), but not in between. We consider first that x'_3 and x'_4 are encountered before z which is visualized in Fig. 9.22(a). Consider the edge e'_5. By (S) it cannot

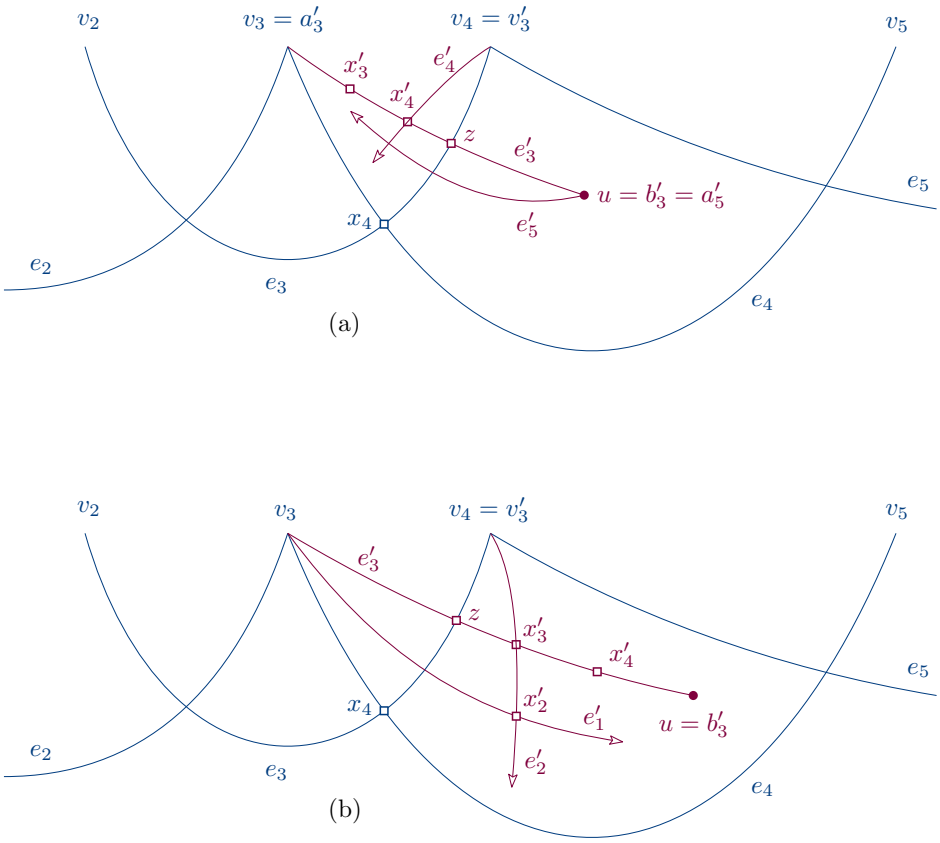


FIGURE 9.22: Proof of Lemma 34.

cross e'_3, and it cannot cross e_4 as otherwise $b'_5 = v_4 = v'_3 = a'_4$ and thus two crossing edges e'_4 and e'_5 would share a common endpoint. Since e'_5 crosses e'_4 on the segment $x'_4b'_4$, which lies inside the region delimited by e'_3, e_4, e_3, it follows that e'_5 crosses e_3, but then the edge is incident to v_3 and we have $e'5 = e'_3$, a contradiction.

It follows that x'_3 and x'_4 are encountered after z as visualized in Fig. 9.22(b), which is exactly the setting covered in Lemma 35. Thus we have that the entire closed path $\mathcal{L}'$ lies in $\mathcal{S}_4$. Consider $x'_2 \in \mathcal{S}_4$. By (S), e'_1 cannot cross e_4 (as $v_3 = a_3 = v'_2 = b'_1$) and it cannot cross e_5 as otherwise e'_1, e_4 and e_5 form (II). Hence e'_1 must cross e_3 and v'_1 is an endpoint of e_3. Since $v_4 = v'_3 \neq v'_1$ holds, we have that $e_3 = (v'_1, v'_3) = e'_2$ and $z = x'_3$ and x'_2 lie on the segment of e_3 between x_4 and x'_3, see Fig. 9.23.

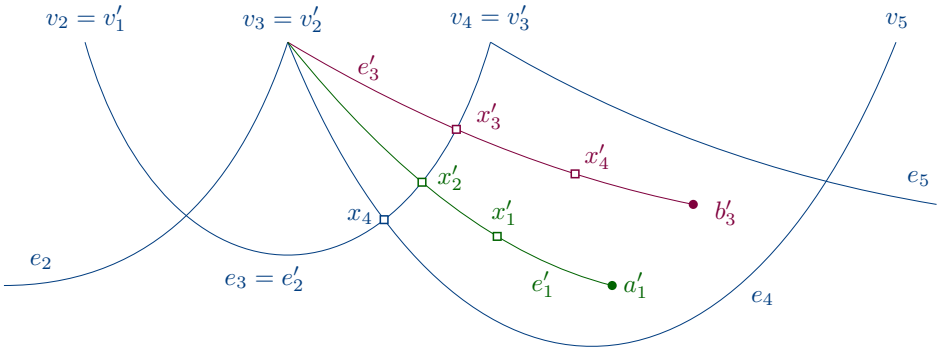


FIGURE 9.23: Proof of Lemma 34, having established $e'_2 = e_3$.

Let us now consider the edges e'_1 and e'_0. As $b'_1 = v_3$ holds, we have that a'_1

is contained inside $\mathcal{S}_4$ by Lemma 29. This also implies that x'_1, which lies in the interior of the segment $a'_1x'_2$, is also inside $\mathcal{S}_4$. Since e'_1 is canonical, we have that e'_0 is incident to $v'_1 = a'_2 = v_2$, but then Lemma 30 ensures implies that e'_0 cannot intersect the interior of $\mathcal{S}_4$, so it cannot contain x'_1, a contradiction. □

We can now finally the state the key lemma.

Lemma 36. *Let C and C' be two chordless odd cycles with ground edges e and e'. If e and e' cross, then e is part of C' and e' is part of C.*

Proof. We will reuse the notation introduced in Lemma 33. In particular, we have $e = e_3$ and $e' = e'_3$. By assumption, e_3 and e'_3 cross in a point z. By Lemma 34, $z \in \hat{e} = \hat{e}_3$. By Lemma 33, z does not lie inside the relative interior of $\hat{e}_3$ - hence z is an endpoint of $\hat{e}_3$, which implies that $e'_3 = e_2$ or $e'_3 = e_4$ and thus $e'_3 \in C$ as desired. Symmetrically, one can show that $e \in C'$. □

We are now ready for the proof of the main theorem.

Proof of Theorem 27. We initialize S to be an empty set. We consider all chordless odd cycles C in $\mathcal{I}$ one after the other. If S does not already contain an edge of C, we add the ground edge of C to S. Otherwise, we skip C.

It is left to show that the set S has the desired properties. By construction, S contains exactly one edge of every chordless odd cycle. Since every odd cycle contains a chordless odd cycle as a subset, (2) follows. It remains to show (1). Suppose for a contradiction that two ground edges e and e' of S intersect. By construction, e and e' are the ground edges of some chordless odd cycle C and C', respectively. W.l.o.g. assume that C was considered before C' when constructing S. By Lemma 36, it follows that $e \in C'$ in which case we would have skipped C' and hence $e' \in S$ is a contradiction to our construction rule which concludes the proof. □

9.4 Open Problems

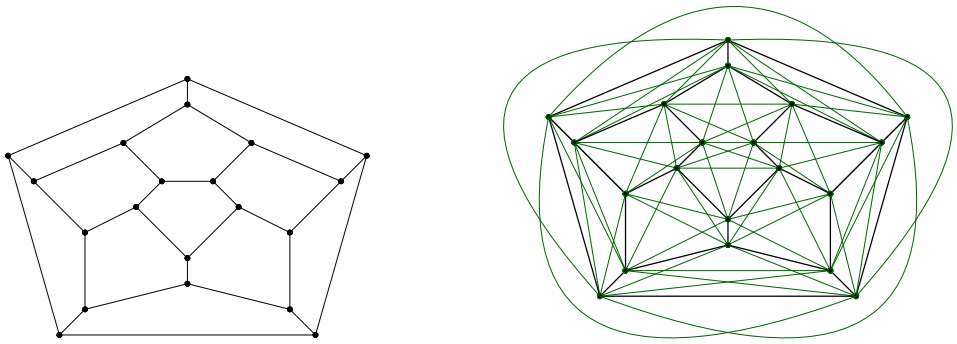
We conclude with some open problems:

1. Can these thickness results be extended to weakly fan-planar graphs? We remark here that already in the characterization of odd chordless cycles, (III) was heavily used - hence such an extension is highly non-trivial.

2. Are there actually strongly fan-planar graphs that have thickness three?

3. Are there strongly fan-planar graphs $\mathcal{G}$ such that *every* fan-planar drawing of $\mathcal{G}$ requires three colors for the edges? In other words, are there strongly fan-planar graphs where odd cycles in the intersection graph of its drawing are unavoidable? Clearly, an affirmative answer to the previous question would imply an affirmative answer to this question.

One potential candidate to answer the last two questions in the affirmative are optimal 2-planar graphs, which have been fully characterized in [29] and which are also fan-planar. It is know that they cannot be decomposed into a 1-plane graph and a forest, while they can be decomposed into a 1-plane graph and a plane graph with maximum degree 12 [25]. However, whether they can be partitioned into two plane graphs is still an open question. At least for some instances, such a decomposition into two plane graphs is in fact possible. Consider the following optimal 2-planar graph with 20 vertices and 90 edges, see Fig. 9.24 for an illustration.



(a) (b)

FIGURE 9.24: (a) The planar skeleton of an optimal 2-planar graph. (b) The full graph, where every face of (a) is completed to a K_5.

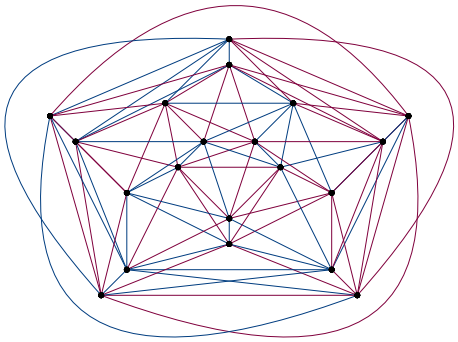
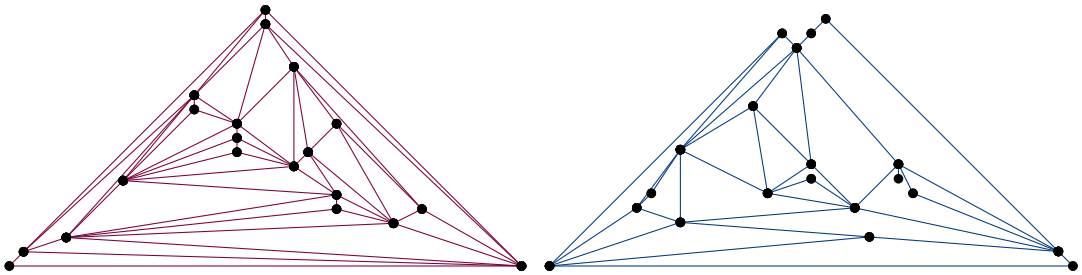


FIGURE 9.25: The edges are partitioned into blue and red edges such that each monochrome subgraph admits a planar drawing.

Fig. 9.25 then shows a partition of the edges such that every induced subgraph is in fact planar, which is witnessed by Fig. 9.26.



(a) (b)

FIGURE 9.26: The two subgraphs are in fact planar.

This partition was found with a rather trivial brute force approach - but maybe there exists an underlying pattern which can be exploited to show that such a partition exists for all optimal 2-planar graphs. If the thickness of the optimal 2-planar graphs turns out to be two, then the following questions would naturally arise.

4. Are there 2-planar graphs of thickness three?

5. If there is no 2-planar graph of thickness three, what is the smallest k such that there exists a k-planar graph of thickness three? Note that K_9 is 4-planar and requires thickness three, hence $k \in \{2, 3, 4\}$.

Part III

Experimental work

Chapter 10

Optimizing multiple aesthetic criteria simultaneously

In Graph Drawing, there is a rich literature and a wide range of techniques for computing drawings of graphs efficiently; see, e.g., [71, 130, 169]. Most of the techniques that have been proposed in the literature over the years exploit structural properties of the input graph to compute a drawing of it such as planarity [97, 110, 126] or degree restriction [27, 33, 168]. To guarantee that the obtained drawings are aesthetically pleasant, a graph drawing algorithm optimizes (over all drawings of the input graph) an objective function that serves as a quality measure for the produced drawing in terms of niceness or legibility; see, e.g., [181]. Supported by empirical and scientific evidences [118, 151, 153], several such quality measures (also called *drawing criteria*) have been proposed and studied over the years.

Some examples are the number of edge crossings [84, 123, 150, 165], the angular resolution[79, 94, 144], the crossing resolution [76, 78], the edge-vertex resolution [28, 56]; see Section 10.1 for formal definitions. However, drawings of graphs that are optimized only in terms of a single criterion may be poor with respect to other criteria and thus may have several undesired properties (e.g., as observed in [24], drawings that optimize the crossing resolution usually contain several adjacent edges that run almost in parallel and vertices that are very close to each other, that is, their angular and edge-vertex resolutions may be arbitrarily bad).

In general, the task of computing a drawing of a graph (especially, when it does not have a special structural property to be exploited) that is optimal in terms of several drawing criteria is difficult and quite challenging. Sometimes it is even impossible, since several criteria "contradict" each other [71]. Hence, it is commonly accepted that aesthetically pleasant drawings of graphs are usually the result of compromising between different aesthetic criteria; see, e.g., [119].

Related work The established techniques that can be used in practice to produce such drawings are usually heuristics. Huang et al. [119] presented a force-directed approach that improves multiple aesthetic criteria (as opposed to standard spring embedding algorithms [81, 99, 142] that usually focus on a single criterion). Davidson and Harel [65] used simulated annealing to produce aesthetically pleasing drawings of small and medium-sized graphs. Even though their approach is easily adaptable to new quality measures in the objective function, it is unfortunately slow (especially when several criteria are taken into account). Devkota et al. [68] presented a stress-based approach, called *stress plus X*, to minimize the number of edge crossings and maximize the crossing resolution, while guaranteeing that the stress of the drawing is also good. Towards a more flexible approach that can handle multiple

drawing criteria, a super set of the authors in [68] presented a gradient descent approach, which can optimize any drawing criterion described or approximated by a differentiable function [2].

Our contribution We generalize the winning approach of the live challenge contests of GD19, GD20 and GD21 to support multiple aesthetic criteria. The algorithmic technique of this approach is based on randomization [116, 147] and on the vertex movement paradigm [37, 98, 154, 176]. Initially, it was introduced to obtain drawings of high crossing resolution [24], while more recently it was adapted to minimize the number of crossings of upwards drawings of graphs and to minimize the edge-length ratio of planar polyline drawings of planar graphs. The algorithmic idea behind this approach is rather simple and intuitive, as it mimics the way a human would try to improve the quality of a drawing. Namely, given a drawing, one would naturally try to identify potential bottlenecks in it (e.g., edges that are involved in several crossings). Having identified such bottlenecks, the next natural step is to move one of the involved vertices (chosen at random) to a new position (chosen at random from a set of randomly generated ones in the neighborhood of the vertex), hoping that this will lead to an improvement.

In this work, we handle any number of drawing criteria (see Figure 10.1 for sample drawings) by introducing a number of extensions to the core of the algorithm. We implemented the extended algorithm and we compared it against the state-of-the-art algorithms mentioned above [2, 65, 119]. Our experimental evaluation shows that our algorithm produces drawings that are, on average, of comparable or of better quality than the other algorithms participating in the experiment.
This chapter is based on (unpublished) joint work with Michael A. Bekos and Patrick Laipple.

10.1 Preliminaries

As already mentioned, we seek to jointly optimize several drawing criteria. We focus on a rich subset, which are commonly used to evaluate the quality of drawings of graphs with up to a few hundred vertices, see [71]. In the following, we introduce each of them together with a brief discussion of different approaches that use them.

Edge crossings. It is commonly accepted and scientifically proved [151] that crossings between edges negatively affect the quality of graph drawings. However, the corresponding crossing-minimization problem is NP-hard [113], and as a result the known heuristics mainly focus on special cases, e.g., as in the Sugiyama framework, where the vertices of the graph are restricted to layers [71, Ch. 9].

Crossing resolution. The negative impact of the edge crossings to the quality of a graph drawing can be mitigated, if the angles formed at the crossings are large [120]. Hence, the problem of maximizing the minimum such angle (called *crossing resolution*) has received considerable attention the past decade, as it is also evident by the fact that it has been the subject of recent graph drawing contests [67, 132]. Since the problem is NP-hard [16], exact approaches mainly focus on instances with specific properties (e.g., bounded degree [13, 15]), while at the same time the corresponding heuristic approaches are significantly fewer [24, 66].

Angular resolution. Another important quality measure of a drawing of a graph, which is one of the oldest considered in the literature [93], is its angular resolution, that is, the minimum angle formed by the edges incident to a vertex; the higher

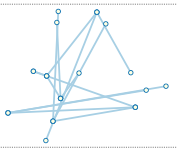
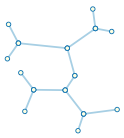
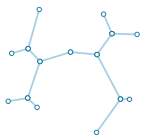
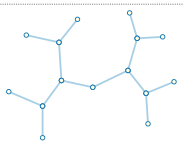
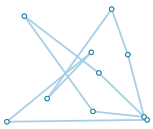
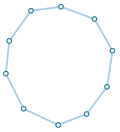
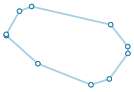
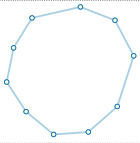
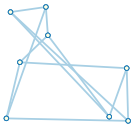
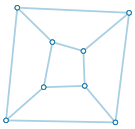
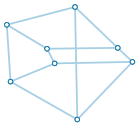
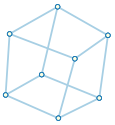
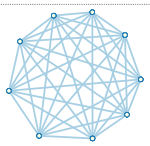
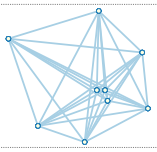
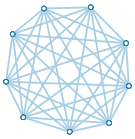
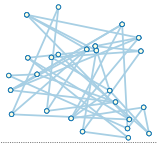
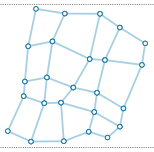
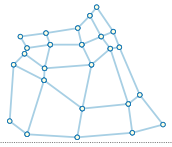
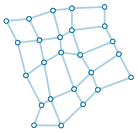
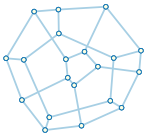
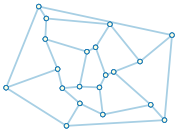
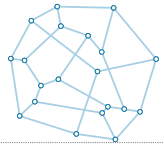
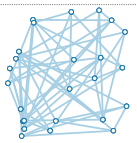
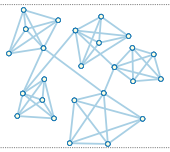
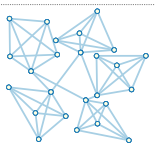
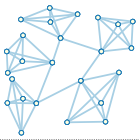
Graph	Random Input	Uniform Weights	Total Resolution	Stress + Length
Tree: $n = 15$ $m = 14$				
Cycle: $n = 10$ $m = 10$				
Cube graph: $n = 8$ $m = 12$				
K_9: $n = 9$ $m = 36$				
5×5 grid: $n = 25$ $m = 40$				
Dodecahedron: $n = 20$ $m = 30$				
Block graph: $n = 25$ $m = 55$				

FIGURE 10.1: Different drawings of common graphs obtained with different parameterizations of our algorithm. Column "Random Input" shows the input drawings. Columns "Uniform Weight", "Total Resolution" and "Stress + Length" show the output drawings. The first takes into account all criteria that we consider with uniform weights. The second is restricted to optimize only the crossing and the angular resolution. The third optimizes stress and ideal edge length.

the angular resolution is, the better the drawing is expected to be. Even though optimizing this parameter turns out to be NP-hard [93], standard spring embedder algorithms [81, 99, 125] tend to result in drawings with good angular resolution due to repulsive forces between pairs of vertices.

Stress and edge-vertex resolution. Standard spring embedder algorithms result in drawings of graphs, which usually have two additional desirable properties; minimal stress and good edge-vertex resolution. Stress is a measure for the difference between vertex-pair distances in a drawing and their graph-theoretic distances (aiming to evaluate how well a drawing captures the graph structure), and can be seen as a special case of a multi-dimensional scaling [137, 164]; early works date back to Kamada and Kawai [125], who formulated stress minimization as an energy optimization problem, and to Gansner et al. [102], who adapted stress majorization instead. On the other hand, the edge-vertex resolution [28, 56] is a measure for the minimum distance between a vertex and an edge in the drawing.

Ideal edge length. While stress captures how well graph distances are realized in a drawing, it does not always prevent the presence of edges with significantly different lengths. Hence we introduce an ideal edge length and we seek for drawings in which the edges have ideally this length. Standard spring embedder algorithms encapsulate this desire by modeling each edge as a spring whose natural length corresponds to its ideal edge length [81]. In other works, the deviation from the ideal edge length is measured by the ratio of the length of the longest edge over the length of the shortest edge in the drawing [42, 139].

10.2 The Drawing Framework

In this section, we present our framework for drawing graphs by taking into account multiple drawing criteria. Our presentation is limited to the criteria of the previous section, but the framework is easily extendable to others. We start with a high-level idea of the algorithmic approach underneath. The input consists of a graph G with n vertices and m edges, an initial drawing Γ_1 of G (e.g., computed at random or by a spring embedder [125]), a set of weights for the different drawing criteria of the objective function ϕ and a bounding box B of width W and height H. At high level, ϕ measures the quality of a drawing by mapping it to a number in $[0, 1]$, such that the better the drawing is, the closer its value to 0 is (see Section 10.2.1). In the output, we seek for a drawing Γ of G contained in B, which ideally is optimal in terms of ϕ (i.e., $\phi(\Gamma) = 0$).

At the i-th iteration of the algorithm, we assume that we have computed a drawing Γ_i of graph G. To compute the next drawing Γ_{i+1} in the iterative approach, we initially select a vertex v of Γ_i from a so-called *vertex-pool*. Intuitively, the vertex-pool contains several vertices that somehow negatively affect the quality of Γ_i with respect to the drawing criteria of ϕ (see Section 10.2.2). Then, we try to identify a new position for vertex v in Γ_i that yields an improved drawing, i.e., $\phi(\Gamma_{i+1}) \leq \phi(\Gamma_i)$. Of course, we cannot consider all possible positions for the vertex v; instead, we consider a small set of randomly generated ones (see Section 10.2.3). Note that in the i-th iteration of our algorithm, we may also decrease the quality of the drawing ($\phi(\Gamma_{i+1}) > \phi(\Gamma_i)$) with a certain probability to avoid locally optimal solutions. As expected, the efficiency of our algorithm depends on the number of drawing criteria that contribute to ϕ, since each requires a number of computations at every iteration of the algorithm. Hence, special care is needed to avoid unnecessary computations, see Section 10.2.4.

10.2.1 The Objective Function

As already mentioned, our drawing framework is easily adaptable to new drawing criteria. This is achieved by maintaining a weighted objective function ϕ, in which the contribution of each drawing criterion is normalized. Formally, let c be a criterion contributing to ϕ, and denote by $c(\Gamma_i)$ the actual value of c in Γ_i, e.g., if c corresponds to the number of crossings and Γ_i contains 10 crossings, then $c(\Gamma_i) = 10$. We define the value $\phi(\Gamma_i)$ of the objective function ϕ for Γ_i as follows:

$$\phi(\Gamma_i) = \sum_{\text{criterion } c} w_c^i \cdot f_c(\Gamma_i)$$

$$\text{where} \qquad f_c(\Gamma_i) = \begin{cases} \frac{c(\Gamma_i)}{norm(c)}, & \text{if c is a criterion to be minimized} \\ 1 - \frac{c(\Gamma_i)}{norm(c)}, & \text{if c is a criterion to be maximized} \end{cases}$$

and $norm(c)$ is an appropriately defined *normalization factor*, such that $f_c(\Gamma_i)$ lies in the interval $[0, 1]$. Regarding the weights w, we set w_c^0 to correspond to the initial weighting of the criteria that is provided as part of the input. Throughout the algorithm, in order to balance out the contribution of the (normalized) criteria, we will adopt the technique introduced in [112], which periodically (in our case, every $x = 100$ iterations) updates the weights based on the average contribution of the criteria to the objective function. Fix a criteria c. Assume that the current iteration is $i \mod x = 0$. In the first step, we will compute the recent (i.e., throughout the last x iterations) rate of change of $c(\Gamma)$ - in practice, we approximate this value by the finite difference approximation $s_c^i = c(\Gamma_i) - c(\Gamma_{i-x})$. Then, our weight will be computed by

$$w_c^i = \frac{e^{\beta s_c^i}}{\sum_{c'} \beta s_{c'}^i} \tag{10.1}$$

where c' ranges over all criteria. The choice of β allows us some kind of flexibility. Namely, for $\beta = 0$ we have an equal weighting of the criteria, if $\beta < 0$, then we favor criteria with a negative rate of change, i.e., we assign more weights to criteria that improved a lot in recent time - conversely, if $\beta > 0$, we favor criteria with a positive rate of change, i.e., we assign more weights to criteria that did not improve (or even decreased) in recent time. Since we aim to compute drawings where every criterion is optimized, we prefer the latter weighting scheme and choose $\beta = 0.1$. We remark here that we also implemented the possibility where the weights of the user, which we denote by w', remain unchanged throughout the algorithm and the actual weights w^* are then obtained by multiplying w' and w (which is computed using the adaptive approach) together - hence, we are free to assign more weight to the choice of the user. Since only the relative values of the weights compared to the other are of interest, we can normalize them such that

$$\sum_{\text{criterion } c} w_c = 1,$$

and thus it follows that the value of the objective function is in the interval $[0, 1]$. Also, by definition of f_c, the closer $\phi(\Gamma_i)$ is to 0, the better Γ_i is. We next describe how the normalization factor of each considered criterion is defined.

Number of crossings. The most natural way to normalize the contribution of the number of crossings to $\phi(\Gamma_i)$ is to set its normalization factor to $\frac{1}{2}m(m-1)$, which is a trivial upper bound on the number of crossings of any drawing of G. However,

in practice, this is inefficient since for drawings containing few crossings but sufficiently many edges this normalization will result in a number close to 0. Therefore, we choose as a normalization factor the maximum of the number of crossings of the initial drawing Γ_1 and of the current drawing Γ_i, that is, we set it to $\max\{\chi(\Gamma_i), \chi(\Gamma_1)\}$, where $\chi(\Gamma_i)$ denotes the number of crossings of Γ_i.

Angular resolution. To normalize the contribution of the angular resolution to $\phi(\Gamma_i)$, we set its normalization factor to $\frac{360^\circ}{\deg(G)}$, where $\deg(G)$ denotes the maximum degree of a vertex of the input graph G.

Crossing resolution. Assuming that Γ_i contains at least one crossing, the contribution of the crossing resolution to $\phi(\Gamma_i)$ is computed by setting its normalization factor to 90°. Otherwise, this criterion will contribute 0 to $\phi(\Gamma_i)$.

Stress. We can normalize the contribution of stress to $\phi(\Gamma_i)$ by setting its normalization factor to $\frac{1}{2}n(n-1)\text{diag}(B)$, where $\text{diag}(B)$ denotes the length of the diagonal of the bounding box B. This approach is reasonable if the Euclidean distance between each pair of vertices is approximately equal to the length of the diagonal of Γ_i, which in practice is highly unlikely. Therefore we substitute $\text{diag}(B)$ with the maximum Euclidean distance between any two vertices in Γ_i.

Edge-vertex resolution. To normalize the contribution of the edge-vertex resolution to $\phi(\Gamma_i)$, we set its normalization factor to the maximum distance between a vertex and an (non-adjacent) edge in Γ_i.

Ideal edge length. Given an ideal edge length L, such that in the output drawing it is desired for all edges to have length L, we measure the contribution of this criterion by the sum of the absolute differences of the edge-lengths and L. To normalize this contribution, we set its normalization factor to $m(\text{diag}(B) - L)$.

10.2.2 The vertex pool

The vertex pool contains several vertices that negatively affect the quality of Γ_i; this idea was introduced in [24] in which the vertex pool consisted of the endvertices of the pair of crossing edges that formed the smallest angle at their crossing. Here, we extend this idea (to support multiple criteria) by introducing a vertex pool, which contains as many *sub-pools* as the number of criteria contributing to ϕ. In particular, each sub-pool p_c contains a set of vertices, called *c-critical*, that negatively affect the quality of Γ_i with respect to a particular criterion c contributing to ϕ. Note that the same vertex may potentially belong to several sub-pools (we provide details on how to determine their content soon).

To compute the next drawing Γ_{i+1}, we need to choose, for some criterion c, a c-critical vertex v from the vertex pool. This is done as follows. We first choose with a certain probability a sub-pool p_c out of the vertex-pool, such that the greater the contribution $w_c \cdot f_c(\Gamma_i)$ of criterion c in ϕ is, the greater the probability to choose sub-pool p_c is. Once p_c is chosen, we choose uniformly at random a vertex from p_c and we set v to be the chosen vertex; note that vertices in p_c may appear with multiplicity, i.e., such vertices are more likely to be chosen.

We are now ready to describe how the content of each sub-pool is determined. This is done in two steps. At high-level, in the first step, we identify for each criterion c that contributes to ϕ, a set of c-critical vertices and we add them to p_c, while in the second step, we enrich the content of p_c with additional vertices that are somehow "close" to the already added vertices. More formally, in the first step we consider each criterion contributing to ϕ and we proceed as follows.

Number of crossings. For an input parameter c_k, we identify the c_k most crossed edges in Γ_i and we add their endpoints to the sub-pool with multiplicities proportional to the number of crossings of their edge. This way, an endpoint of an edge involved in several crossings is more likely to be selected.

Angular resolution. We identify the vertex u and its two incident edges e_1 and e_2 defining the angular resolution of Γ_i. Then, we add u as well as the endpoints of e_1 and e_2 different from u to the sub-pool. To increase the probability of selecting u, we assume that it appears in the sub-pool with a multiplicity of 3.

Crossing resolution. As above, we identify the two edges, which define the crossing resolution of Γ_i (if any), and we add their endpoints to the sub-pool.

Stress. For an input parameter s_k, we identify the s_k vertices that contribute the most to the stress of Γ_i and we add them to the sub-pool.

Edge-vertex resolution. We identify the vertex u and the edge e that define the edge-vertex resolution in Γ_i. We add u and the endpoints of e to the sub-pool.

Ideal edge length. We identify the shortest and longest edge of Γ_i and we add their endpoints to the sub-pool. The idea of the second step is that one may improve the quality of Γ_i, if first the location of a vertex in the neighborhood of a c-critical vertex is changed. The algorithmic implementation of this idea is achieved by enriching each sub-pool with vertices that are neighboring the ones already in it, where the neighborhood can be defined either in terms of actual geometry (i.e., Euclidean distance), or combinatorially (i.e., graph-theoretic distance). We adopted the latter approach for efficiency reasons. In our probabilistic selection procedure, the closer a vertex is to the initial content of the sub-pool, the more likely it is to be chosen.

A small caveat. Empirical evaluations suggest that for a subset of the criteria, the computational overhead of constructing the vertex pools might worsen the performance in comparison to a random choice. In particular, we observed this behavior for so called *global* criteria c, where the aggregation function of $c(\Gamma)$ is part of $\{SUM, AVG\}$ instead of $\{MIN, MAX\}$. In particular, for our choice of criteria, this is the case for the crossing number, the stress and the ideal edge length. Hence, we adopted the following strategy for step i. If i is even, then we chose a random vertex to move. Otherwise, if i is odd, then we compute the vertex-pool for the local (i.e., none global) criteria and choose our vertex accordingly. Clearly, instead of alternating every iteration, we could make this choice dependent on the contribution of the local and global criteria to the loss function, their current weighting or a combination of the two.

10.2.3 Computing the next drawing

Let v be the c-critical vertex of drawing Γ_i that has been chosen from sub-pool p_c at the i-th iteration of our algorithm. To compute the position of v in the next drawing Γ_{i+1}, we adopt the successful approach of [24] with a critical modification for reasons of efficiency. In particular, as in [24], for an input parameter $\rho > 0$, we consider a set of ρ equispaced rays that all emanate from v in Γ_i, such that the angle formed by the first ray and by the horizontal axis is chosen uniformly at random from the interval $[0, 2\pi]$. The idea is to introduce a sample position along each of these rays that is in some distance from v which is specified using randomization again; sampled positions outside the bounding box $\mathcal{B}$ are disregarded. However, unlike in [24], we do not necessarily process each of these sampled positions. Instead, we process them in a random order and we move v to the first encountered position that improves the

quality of Γ_i (if any). This yields the next drawing Γ_{i+1}, for which $\phi(\Gamma_{i+1}) \leq \phi(\Gamma_i)$ holds.

On the other hand, if none of the sampled positions yields an improved drawing, then the algorithm may reduce the quality of Γ_i with a certain probability, which is a common approach for avoiding local optimal solutions. Inspired by standard techniques from simulated annealing, the algorithm may move vertex v at the last sampled position (for which $\phi(\Gamma_{i+1}) > \phi(\Gamma_i)$ holds) with probability $e^{-c/(t+1)}$, where c is a small constant and t is the number of iterations performed by the algorithm without updating the drawing, e.g., for $c = 10$ and $t = 9$, the probability to reduce the quality of the drawing is $\frac{1}{e} \approx 0.368$.

10.2.4 Further Insights to the Implementation

As already mentioned in the previous part, the efficiency of our approach highly depends on the number of criteria that contribute to the objective function (since each requires a number of computations) and on the approach that one adopts in order to compute the content of each sub-pool (i.e., the c-critical vertices for each criterion c) at every iteration of the algorithm. At the start of our algorithm, the crucial (and unavoidable) part is to determine the value of the objective function, which naïvely needs $\mathcal{O}(m^2)$ time due to the computation of crossings. At every iteration, we avoid determining from scratch the value of the objective function by maintaining additional information. By doing so, this, as well as the content of each sub-pool, can be computed in time $\mathcal{O}(deg(v) \cdot m)$, where v is the vertex that is moved. To make this more evident, we describe at a higher level the necessary actions that need to be performed both at the initialization of our algorithm and at each subsequent iteration of it. In the former, the crucial part is to determine the value of the objective function, which is equivalent to determining the value $f_c(\Gamma_1)$ of each criterion c in the initial drawing Γ_1. This step may be inevitably demanding. In particular, the bottleneck of this approach is usually related to determining the crossings of the drawing, since a naïve approach may require $\mathcal{O}(m^2)$ time, which is a worst-case tight bound. In practice, however, it is too unlikely for drawing Γ_1 (and for subsequent drawings) to have a quadratic number of crossings. With this in mind, we adopted in our implementation a data structure, called R-tree [30], which stores for each edge e a minimum-area rectangle $R(e)$ bounding e. To determine the number of crossings of edge e using this data structure, we first report the rectangles overlapping $R(e)$, which can be efficiently done in time logarithmic in the size of the data structure and linear in the size of the output. Then, the edges crossing e form a subset of the edges associated to the reported rectangles. Even though, in worst case, this approach is not asymptotically better than the naïve one, in practice, it turns out to be more efficient.

At iteration i of our algorithm, it is crucial to avoid determining from scratch the value $f_c(\Gamma_i)$ of each criterion c in Γ_i (since this is equally demanding as the initialization of the algorithm). To achieve this and to efficiently compute the c-critical vertices of each criterion c, we store additional information (which we need to keep updated throughout the algorithm):

(i) for each vertex, we store the smallest angle formed by its incident edges, as well as the corresponding pair of edges,

(ii) for each edge, we store its length, the number of edges crossing it is part of and the corresponding smallest crossing angle,

(iii) for each pair of a vertex and a non-incident edge and for each pair of vertices, we store their Euclidean distance and

(iv) for each pair of edges, we store whether they cross and the corresponding crossing angle.

Since Γ_{i-1} and Γ_i differ by the position of a single vertex v, maintaining/updating this information can be done in time $O(\deg(v) \cdot m)$. The real benefit of this approach is that the c-critical vertices for each criterion c of ϕ, as well as the value of the objective function for each sample position considered for vertex v can be computed in $O(n + m)$ time. Hence, a single iteration of our algorithm needs $O(\deg(v) \cdot m)$, which, in practice, turns out to be more efficient than the naïve $O(m^2)$.

10.3 Experimental Evaluation

In this section, we describe the experiment that we conducted and present the result of our evaluation. In our experiment, we compared our algorithm against the following algorithms:

(A) the simulated annealing based algorithm by Davidson and Harel [65],

(B) the Bigangle algorithm by Eades et al. [119], and

(C) the gradient-descent based algorithm $(GD)^2$ by Ahmed et al. [2].

To ensure that the comparison between the algorithms is fair, besides our algorithm, we also implemented all aforementioned algorithms in Python. Our implementation of the Bigangle algorithm is an extension of the standard spring embedder algorithm by Fruchterman and Reingold [99], whose implementation we adopted from the NetworkX [111] library. As a test set for our experiment, we used the Rome graphs [70], which are roughly 11.500 graphs, whose number of vertices ranges from 10 to 100. In this regard, we decided to set the size of the bounding box B containing the output drawing to 100×100. We remark that even though many graphs that arise nowadays may be several magnitudes larger than the ones of our test set, the methodology to draw graphs as a node-link diagram as well as the drawing criteria introduced in this work do not translate well to such sizes. The results that we report in this section are on average across different drawings with the same number of vertices. The experiment was performed as follows: For every graph of the test set, we computed an initial drawing which was the input to all algorithms of the experiment. This initial drawing was computed using the spring embedder algorithm by Kamada and Kawai [125] that is part of the NetworkX library. To ensure a fair comparison, we set a limit for the time needed by each algorithm to compute a drawing of each graph of the test set that is dependent on the number n of its vertices. This time-limit was set to the maximum of 3 and $\frac{n}{10}$ seconds. However, since the cooling schedule of the Bigangle algorithm (that was inherited from the one of Fruchterman and Reingold) requires to know the number of iterations a priori, we set it to 50, which we observed to comply with our time-limit, as for medium-size graphs ($n \approx 50$), the 50 iterations needed roughly 5 seconds (note that also the original algorithm by Fruchterman and Reingold was observed to converge after 50 iterations [99]).

10.3.1 Parameter choice and normalization

Since we sought to evaluate the algorithms participating in our experiment in terms of different aesthetic criteria, we tried to weight equally the contribution of each criterion to the objective function of each algorithm. Since in our algorithm the contribution of each criterion is already normalized, we simply set its weight factor to 1 over the number of criteria. This task was not as easy for the other algorithms of the experiment, and we next describe the encountered problems.

Davidson and Harel's algorithm. In the original paper [65], the authors do not describe how to normalize the contribution of the different criteria to the objective function. Thus, we had to average and weight them by empirically found constants to ensure that they are considered as "almost equal".

Bigangle. In the original paper [119], the constants of the magnitudes of the force for the crossing and for the angular resolution were not specified. We observed that it was beneficial to dynamically set them to approximately the same magnitude as the node-repulsion and the edge-attraction forces in every iteration.

$(GD)^2$. As in the work by Davidson and Harel, the authors of [2] do not describe how to normalize the contribution of the different criteria to the objective function. Instead, in their prototype the user is responsible to find a good weighting scheme for the criteria. To include this algorithm in our experiment, we had to adopt a similar approach as for the algorithm by Davidson and Harel.

10.3.2 Our findings

We are now ready to discuss the actual results of the experimental evaluation; for a summary refer to Figure 10.2, in which each curve illustrates the average values across different drawings with the same number of vertices.

Angular and crossing resolution. Our results for the angular and the crossing resolution are given in Figs. 10.2a and 10.2b. It is immediate to see that our algorithm outperforms the remaining algorithms of the experiment significantly. Up to a certain degree, this was expected since our framework builds upon the algorithm of [24], which was explicitly configured to optimize the crossing resolution. In the multicriteria setting, we remark that the performance of the algorithm (both in terms of angular and crossing resolution) is still good. For the remaining algorithms of the experiment, we observe that $(GD)^2$ yields drawings of better angular and crossing resolution than the Bigangle and the one of Davidson and Harel for small graphs, while this advantage seems to diminish for larger ones.

Number of crossings. Our results are shown in Fig. 10.2c. Our algorithm yields, together with the one of Kamada and Kawai [125], the least number of crossings. Here, Bigangle yields by far the worst solution, while the one of Davidson and Harel performs third best.

Stress. As the algorithm by Kamada and Kawai [125], which we used to generate the input drawings for each algorithm of the experiment, has been designed to optimize stress, we decided to include its performance in all categories, in particular in Fig. 10.2d. Here, the interesting observation is that our algorithm produces drawigs of comparable stress to the one of Kamada and Kawai, even though we optimize for several criteria at the same time. $(GD)^2$ also produces good results regarding the stress, which could maybe be attributed to the fact that the stress function is easily

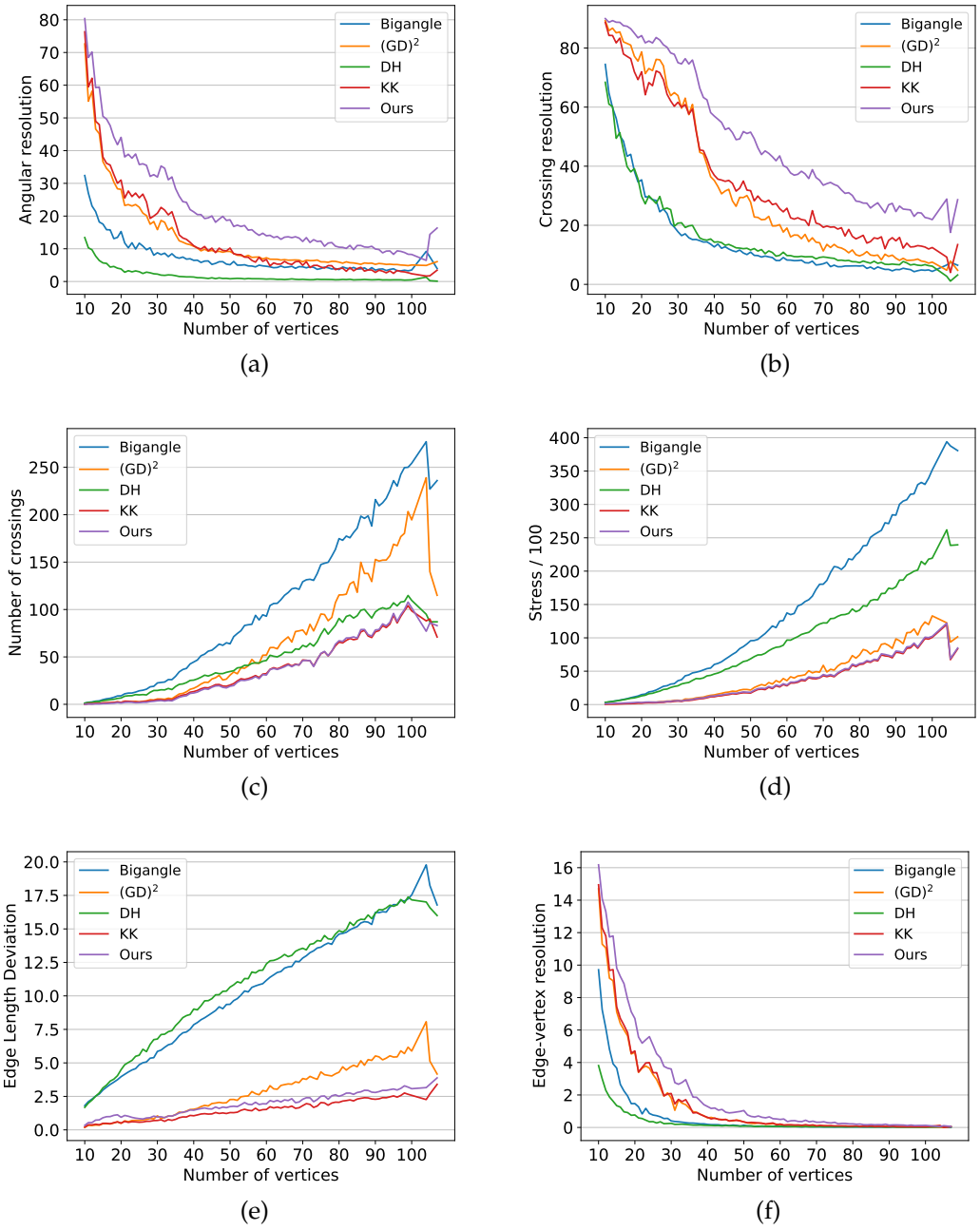


FIGURE 10.2: The results of our experimental evaluation for the Rome graphs.

differentiable and does not require an surrogate loss function. It is interesting to observe that even though the Bigangle algorithm extends a spring embedder, which is in practice used to minimize the stress, its performance is the worst over all - the issue here is that the attractive and repulsive forces are very sensitive to the choice of correct parameters, which might be the reason for its poor performance.

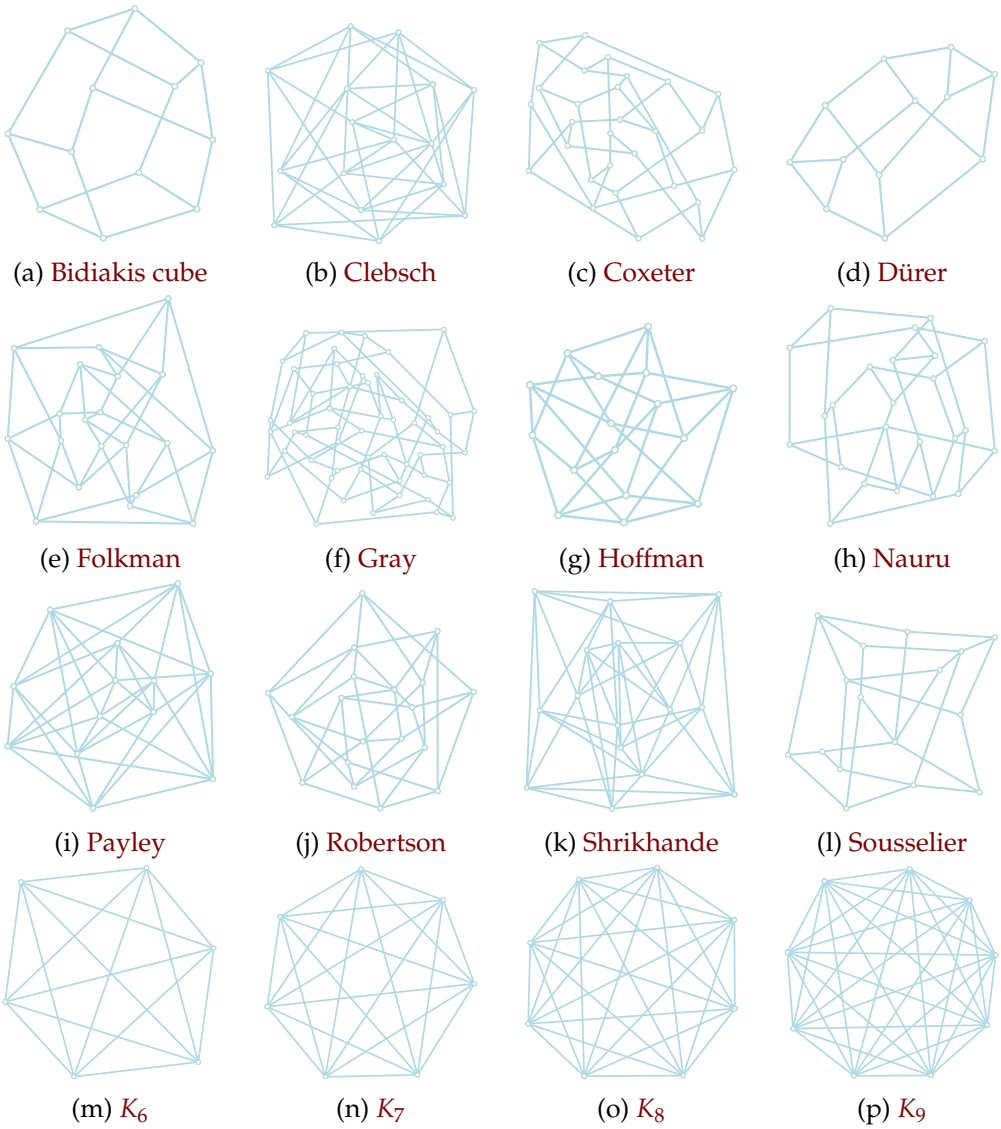
Ideal edge length. Since it is difficult to define a reasonable value for the ideal edge length for graphs of different sizes, we followed the approach of [125] in our experiment, which defines the ideal length of an edge to be $\frac{\min\{W,H\}}{\text{diam}(G)}$, where W and H are the width and the height of the bounding box B, respectively, and $\text{diam}(G)$

is the graph diameter of G. In Fig. 10.2e, we plot the deviation from this length on average. We remark here that this is the only criterion where our algorithm is not best-possible, which is to be expected as the algorithm of [125] explicitly enforces this edge-length. However, the difference is quite small and hence a good trade-off when compared to the performance of the other criteria.

Edge-vertex resolution. The results for the edge-vertex resolution are very comparable to those for the angular resolution, see Fig. 10.2f

10.4 Sample Drawings

In this section, we present sample drawings of well-known graphs produced by our algorithm, see Fig. 10.3. Here, we observe that weighting each criterion equally (which follows by our choice of the adaptive weight scheme) does not necessarily yield a drawing of good quality (as it is clear that there is room for improvement), especially for graphs that are not symmetric. The achieved values for each criterion are reported in Table 10.1.



(a) Bidiakis cube (b) Clebsch (c) Coxeter (d) Dürer

(e) Folkman (f) Gray (g) Hoffman (h) Nauru

(i) Payley (j) Robertson (k) Shrikhande (l) Sousselier

(m) K_6 (n) K_7 (o) K_8 (p) K_9

FIGURE 10.3: Sample drawings.

Graph	n	m	Angular Res.	Crossing Res.	# Crossings	Edge Length	Stress	Vertex Res.
Bidiakis	12	18	44°	89°	3	27	456	11
Clebsch	16	40	6°	38°	63	37	2040	0.09
Coxeter	28	42	31°	67°	19	17	5067	0.9
Dürer	12	18	55°	89°	2	19	209	7
Folkman	20	40	13°	72°	17	15	1402	3
Gray	54	81	14°	47°	72	8	14451	0.04
Hoffman	16	32	20°	58°	24	19	762	4
Nauru	24	36	34°	59°	18	17	3220	0.6
Payley	13	39	9°	48°	42	32	1071	1.3
Robertson	19	38	14°	46°	35	25	1599	0.2
Shrikhande	16	48	5°	40°	55	34	1715	0.2
Sousselier	16	27	23°	74°	11	24	888	4.6
K_6	6	15	22°	56°	15	47	128	21
K_7	7	21	24°	48°	35	38	243	17
K_8	8	28	18°	40°	35	36	344	12
K_9	9	45	15°	36°	126	34	484	9

TABLE 10.1: The values of each criterion for the drawings of Fig. 10.3

10.5 Open problems

We presented an algorithmic framework that incorporates multiple aesthetic criteria to draw graphs. Our experimental evaluation implies that our approach outperforms other multi-criteria approaches from the literature in all considered criteria, while it achieves comparable performance to single criteria optimizations. Notably, the framework is easily extendable to new criteria, as each new criterion requires "only" the definition of a normalization function and the classification into local or global criteria, which usually directly follows from the criteria definition. Further, we can also extend our algorithmic framework to other drawing constraints, such as the presence of an underlying grid (which can be achieved by rounding the coordinates to grid points) or specific directions for the edges (e.g., upwardness, which can be achieved by adapting the sampling scheme of new point locations). Another use case is for example to find drawings with low local crossing number. In the following, we pose open problems raised by our work.

1. As already mentioned, we used the algorithm by Kamada and Kawai for the input drawings. This decision was made in order to speed up the experiment, since starting from randomly generated drawings (as, e.g., those of Fig. 10.1) would require higher computational time to reach drawings of comparable quality. With the time-limit that we set for our experiment, a better input usually led to a better output for all algorithms of the experiment. Hence, the question arises whether we can find fast algorithms that attain a decent quality in many considered criteria?

2. Related to the presence of a time-limit, it is interesting to study the quality of the drawings when no such restriction is imposed. Is there a criterion that is not too time consuming but still admits (almost) convergence?

3. The iterative nature of the algorithm gives the impression that it has the potential to be parallelized in order to handle even larger graphs. However, several aspects (also depending on the considered criteria) need attention, especially when we seek to improve the quality of the drawing at each iteration.

4. While the vertex movement paradigm is ideal for small graphs, one may consider moving larger subgraphs (e.g., small cliques) for larger graphs. In particular, one interesting idea would be to optimize locally dense subgraphs separately and in a next step join these results together.

5. As already mentioned, our algorithm is tailored to graphs of small to medium size. This is justified both by the time complexity (which is in the worst case quadratic in the number of edges per iteration) and by the fact that taking into account several criteria simultaneously is on its own a difficult problem. Identifying appropriate criteria for larger graphs and developing an efficient framework to cope with them is a challenging task for future considerations.

Chapter 11

Conclusions

In Chapter 1, we considered two sets of extensions to planar graphs - whereas one set arises naturally based on the properties that planar graphs admit, the other one follows from cognitive experiments regarding the readability of a drawing. We covered relevant notation and preliminaries in Chapter 2 and Chapter 3, before we considered three of the most researched topics of planar graphs for some of their extensions in part one. In particular, we considered the recognition problem for k-planar graphs in Chapter 4 and provided both exact and approximate solutions for a restricted class of input graphs. In Chapter 5, we provided the first morphing algorithm that can handle a (subset) of graphs of non-constant genus. In Chapter 6 we provide constructive algorithms that certify the containment of low-degree graphs in the class of k-bend RAC graphs for small k. In the second part, we considered structural properties of beyond-planar graph classes. In Chapter 7 we introduced a new class called apRAC graphs, where we study edge-density results, containment relationships and the hardness of the recognition problem. In Chapter 8, we establish an upper-bound on the edge-density of bipartite gap-planar graphs. We conclude this part by a study of fan-planar graphs in Chapter 9, where we first show that the required definition of fan-planarity in terms of three forbidden crossing configurations is in fact not superfluous, as the removal of any such configuration yields a proper superclass - in the latter part, we study the maximum thickness that a fan-planar graph can have. Finally, in the last part of the thesis, we provide in Chapter 10 an algorithmic framework which jointly optimizes several aesthetic criteria. The framework is easily extendable to additional criteria and the empirical performance suggests its worth.

At the end of every chapter, we stated related open-problems to the content that was covered.

Bibliography

[1] Ackerman, E.: On topological graphs with at most four crossings per edge. Comput. Geom. 85 (2019)

[2] Ahmed, A.R., Luca, F.D., Devkota, S., Kobourov, S.G., Li, M.: Graph drawing via gradient descent, $(gd)^2$. In: GD. Lecture Notes in Computer Science, vol. 12590, pp. 3–17. Springer (2020)

[3] Ajtai, M., Chvátal, V., Newborn, M., Szemerédi, E.: Crossing-free subgraphs. In: Hammer, P.L., Rosa, A., Sabidussi, G., Turgeon, J. (eds.) Theory and Practice of Combinatorics, North-Holland Mathematics Studies, vol. 60, pp. 9–12. North-Holland (1982)

[4] Alamdari, S., Angelini, P., Barrera-Cruz, F., Chan, T.M., Da Lozzo, G., Di Battista, G., Frati, F., Haxell, P., Lubiw, A., Patrignani, M., Roselli, V., Singla, S., Wilkinson, B.T.: How to morph planar graph drawings. SIAM J. Computing 46(2), 824–852 (2017)

[5] Albertson, M.O.: Chromatic number, independence ratio, and crossing number. Ars Math. Contemp. 1(1) (2008)

[6] Alexa, M., Cohen-Or, D., Levin, D.: As-rigid-as-possible shape interpolation. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 157–164 (2000)

[7] Angelini, P., Bekos, M.A., Förster, H., Kaufmann, M.: On RAC drawings of graphs with one bend per edge. Theor. Comput. Sci. 828-829, 42–54 (2020), https://doi.org/10.1016/j.tcs.2020.04.018

[8] Angelini, P., Bekos, M.A., Katheder, J., Kaufmann, M., Pfister, M.: RAC drawings of graphs with low degree. In: MFCS. LIPIcs, vol. 241, pp. 11:1–11:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022)

[9] Angelini, P., Bekos, M.A., Katheder, J., Kaufmann, M., Pfister, M.: Axis-parallel right angle crossing graphs. In: 39th European Workshop on Computational Geometry (EuroCG'23) (2023)

[10] Angelini, P., Bekos, M.A., Kaufmann, M., Pfister, M., Ueckerdt, T.: Beyond-planarity: Turán-type results for non-planar bipartite graphs. In: ISAAC. LIPIcs, vol. 123, pp. 28:1–28:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018)

[11] Angelini, P., Bekos, M.A., Kaufmann, M., Schneck, T.: Efficient generation of different topological representations of graphs beyond-planarity. In: International Symposium on Graph Drawing and Network Visualization. pp. 253–267. Springer (2019)

[12] Angelini, P., Bekos, M.A., Montecchiani, F., Pfister, M.: On morphs of 1-plane graphs. J. Comput. Geom. 13(1) (2022), https://doi.org/10.20382/jocg.v13i1a10

[13] Angelini, P., Cittadini, L., Didimo, W., Frati, F., Di Battista, G., Kaufmann, M., Symvonis, A.: On the perspectives opened by right angle crossing drawings. J. Graph Algorithms Appl. 15(1), 53–78 (2011)

[14] Angelini, P., Da Lozzo, G., Frati, F., Lubiw, A., Patrignani, M., Roselli, V.: Optimal morphs of convex drawings. In: Arge, L., Pach, J. (eds.) Symposium on Computational Geometry (SoCG 2015). LIPIcs, vol. 34, pp. 126–140. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015)

[15] Argyriou, E.N., Bekos, M.A., Kaufmann, M., Symvonis, A.: Geometric RAC simultaneous drawings of graphs. J. Graph Algorithms Appl. 17(1), 11–34 (2013)

[16] Argyriou, E.N., Bekos, M.A., Symvonis, A.: The straight-line RAC drawing problem is NP-hard. J. Graph Algorithms Appl. 16(2), 569–597 (2012)

[17] Arikushi, K., Fulek, R., Keszegh, B., Moric, F., Tóth, C.D.: Graphs that admit right angle crossing drawings. Comput. Geom. 45(4), 169–177 (2012)

[18] Aronov, B., Seidel, R., Souvaine, D.: On compatible triangulations of simple polygons. Comput. Geom. Theory Appl. 3(1), 27–35 (1993)

[19] Auer, C., Bachmaier, C., Brandenburg, F.J., Gleißner, A., Hanauer, K., Neuwirth, D., Reislhuber, J.: Outer 1-planar graphs. Algorithmica 74(4), 1293–1320 (2016)

[20] Auer, C., Brandenburg, F.J., Gleißner, A., Reislhuber, J.: 1-planarity of graphs with a rotation system. J. Graph Algorithms Appl. 19(1), 67–86 (2015)

[21] Bae, S.W., Baffier, J., Chun, J., Eades, P., Eickmeyer, K., Grilli, L., Hong, S., Korman, M., Montecchiani, F., Rutter, I., Tóth, C.D.: Gap-planar graphs. Theor. Comput. Sci. 745, 36–52 (2018)

[22] Bannister, M.J., Cabello, S., Eppstein, D.: Parameterized complexity of 1-planarity. In: Proceedings of the 13th International Conference on Algorithms and Data Structures. p. 97–108. WADS'13, Springer-Verlag, Berlin, Heidelberg (2013), https://doi.org/10.1007/978-3-642-40104-6_9

[23] Batini, C., Furlani, L., Nardelli, E.: What is a good diagram? A pragmatic approach. In: ER. pp. 312–319. IEEE Computer Society and North-Holland (1985)

[24] Bekos, M.A., Förster, H., Geckeler, C., Holländer, L., Kaufmann, M., Spallek, A.M., Splett, J.: A heuristic approach towards drawings of graphs with high crossing resolution. Comput. J. 64(1), 7–26 (2021), https://doi.org/10.1093/comjnl/bxz133

[25] Bekos, M.A., Giacomo, E.D., Didimo, W., Liotta, G., Montecchiani, F., Raftopoulou, C.N.: Edge partitions of optimal 2-plane and 3-plane graphs. In: WG. Lecture Notes in Computer Science, vol. 11159, pp. 27–39. Springer (2018)

[26] Bekos, M.A., Grilli, L.: Fan-planar graphs. In: Hong, S.H., Tokuyama, T. (eds.) Beyond Planar Graphs: Communications of NII Shonan Meetings, pp. 131–148. Springer (2020)

[27] Bekos, M.A., Gronemann, M., Kaufmann, M., Krug, R.: Planar octilinear drawings with one bend per edge. J. Graph Algorithms Appl. 19(2), 657–680 (2015), https://doi.org/10.7155/jgaa.00369

[28] Bekos, M.A., Gronemann, M., Montecchiani, F., Symvonis, A., Theocharous, L.: Grid drawings of graphs with constant edge-vertex resolution. Computational Geometry 98 (2021), https://www.sciencedirect.com/science/article/pii/S0925772121000456

[29] Bekos, M.A., Kaufmann, M., Raftopoulou, C.N.: On Optimal 2- and 3-Planar Graphs. In: 33rd International Symposium on Computational Geometry (SoCG 2017). LIPIcs, vol. 77, pp. 16:1–16:16 (2017)

[30] Bentley, J.L.: Decomposable searching problems. Inf. Process. Lett. 8(5), 244–251 (1979), https://doi.org/10.1016/0020-0190(79)90117-0

[31] de Berg, M., Cheong, O., van Kreveld, M.J., Overmars, M.H.: Computational geometry: algorithms and applications, 3rd Edition. Springer (2008)

[32] Biedl, T., Lubiw, A., Petrick, M., Spriggs, M.: Morphing orthogonal planar graph drawings. ACM Trans. Algorithms 9(4), 29:1–29:24 (2013)

[33] Biedl, T.C., Kant, G.: A better heuristic for orthogonal graph drawings. Comput. Geom. 9(3), 159–180 (1998), https://doi.org/10.1016/S0925-7721(97)00026-6

[34] Bienstock, D., Dean, N.: Bounds for rectilinear crossing numbers. J. Graph Theory 17(3), 333–348 (1993), https://doi.org/10.1002/jgt.3190170308

[35] Binucci, C., Di Giacomo, E., Didimo, W., Montecchiani, F., Patrignani, M., Symvonis, A., Tollis, I.G.: Fan-planarity: Properties and complexity. Theoretical Computer Science 589, 76–86 (2015), https://www.sciencedirect.com/science/article/pii/S0304397515003515

[36] Bläsius, T., Kobourov, S.G., Rutter, I.: Simultaneous embedding of planar graphs. CoRR abs/1204.5853 (2012), http://arxiv.org/abs/1204.5853

[37] Bläsius, T., Radermacher, M., Rutter, I.: How to draw a planarization. In: Steffen, B., Baier, C., van den Brand, M., Eder, J., Hinchey, M., Margaria, T. (eds.) SOFSEM. LNCS, vol. 10139, pp. 295–308. Springer (2017), https://doi.org/10.1007/978-3-319-51963-0_23

[38] Bodendiek, R., Schumacher, H., Wagner, K.: Bemerkungen zu einem Sechsfarbenproblem von G. Ringel. Abhandlungen aus dem Mathematischen Seminar der Universitaet Hamburg 53(1), 41–52 (1983)

[39] Bodendiek, R., Schumacher, H., Wagner, K.: Über 1-optimale graphen. Mathematische Nachrichten 117, 323 – 339 (02 2009)

[40] Bodlaender, H.L., Kloks, T.: Efficient and constructive algorithms for the pathwidth and treewidth of graphs. Journal of Algorithms 21(2), 358–402 (1996), https://www.sciencedirect.com/science/article/pii/S0196677496900498

[41] Bondy, J., Murty, U.: Graph Theory. Springer Publishing Company, Incorporated, 1st edn. (2008)

[42] Borrazzo, M., Frati, F.: On the planar edge-length ratio of planar graphs. J. Comput. Geom. 11(1), 137–155 (2020), https://journals.carleton.ca/jocg/index.php/jocg/article/view/470

[43] Brandenburg, F.J.: Recognizing optimal 1-planar graphs in linear time. Algorithmica 80(1), 1–28 (2018)

[44] Brandenburg, F.J.: Characterizing and recognizing 4-map graphs. Algorithmica 81(5), 1818–1843 (2019)

[45] Brandenburg, F.J.: On fan-crossing graphs. Theor. Comput. Sci. 841, 39–49 (2020), https://doi.org/10.1016/j.tcs.2020.07.002

[46] Brandenburg, F.J., Didimo, W., Evans, W.S., Kindermann, P., Liotta, G., Montecchiani, F.: Recognizing and drawing IC-planar graphs. Theor. Comput. Sci. 636, 1–16 (2016)

[47] Brinkmann, G., Greenberg, S., Greenhill, C.S., McKay, B.D., Thomas, R., Wollan, P.: Generation of simple quadrangulations of the sphere. Discret. Math. 305(1-3), 33–54 (2005), https://doi.org/10.1016/j.disc.2005.10.005

[48] Bruckdorfer, T., Cornelsen, S., Gutwenger, C., Kaufmann, M., Montecchiani, F., Nöllenburg, M., Wolff, A.: Progress on partial edge drawings. CoRR abs/1209.0830 (2012)

[49] C., H.S.M.: Vorlesungen über die Theorie der Polyeder. By E. Steinitz. Die Grundlehren der mathematischen Wissenschaften, 41. 1934. (J. Springer, Berlin), vol. 19. Cambridge University Press (1935)

[50] Cabello, S.: Hardness of approximation for crossing number. Discrete and Computational Geometry 49 (04 2012)

[51] Cairns, S.S.: Deformations of plane rectilinear complexes. American Math. Monthly 51(5), 247–252 (1944)

[52] Chambers, E.W., Erickson, J., Lin, P., Parsa, S.: How to morph graphs on the torus. In: ACM-SIAM Symposium on Discrete Algorithms (SODA 2021). pp. 2759–2778. SIAM (2021)

[53] Chaplick, S., van Dijk, T.C., Kryven, M., Park, J., Ravsky, A., Wolff, A.: Bundled crossings revisited. Journal of Graph Algorithms and Applications 24(4), 621–655 (2020)

[54] Cheong, O., Pfister, M., Schlipf, L.: The thickness of fan-planar graphs is at most three. In: GD. Lecture Notes in Computer Science, vol. 13764, pp. 247–260. Springer (2022)

[55] Chiba, N., Yamanouchi, T., Nishizeki, T.: Linear algorithms for convex drawings of planar graphs. Progress in graph theory 173, 153–173 (1984)

[56] Chrobak, M., Goodrich, M.T., Tamassia, R.: Convex drawings of graphs in two and three dimensions (preliminary version). In: Whitesides, S. (ed.) Proceedings of the Twelfth Annual Symposium on Computational Geometry, Philadelphia, PA, USA, May 24-26, 1996. pp. 319–328. ACM (1996), https://doi.org/10.1145/237218.237401

[57] Chuzhoy, J., Mahabadi, S., Tan, Z.: Towards better approximation of graph crossing number. In: 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS). pp. 73–84. IEEE (2020)

[58] Cornelsen, S., Pfister, M., Förster, H., Gronemann, M., Hoffmann, M., Kobourov, S.G., Schneck, T.: Drawing shortest paths in geodetic graphs. J. Graph Algorithms Appl. 26(3), 353–361 (2022)

[59] Courcelle, B.: The monadic second-order logic of graphs. i. recognizable sets of finite graphs. Information and Computation 85(1), 12–75 (1990), https://www.sciencedirect.com/science/article/pii/089054019090043H

[60] Courcelle, B.: The monadic second-order logic of graphs xii: planar graphs and planar maps. Theoretical Computer Science 237(1), 1–32 (2000), https://www.sciencedirect.com/science/article/pii/S0304397599003059

[61] Courcelle, B.: The monadic second-order logic of graphs xiv: uniformly sparse graphs and edge set quantifications. Theoretical Computer Science 299(1), 1–36 (2003), https://www.sciencedirect.com/science/article/pii/S0304397502005789

[62] Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015)

[63] Czap, J., Šugerek, P.: Drawing graph joins in the plane with restrictions on crossings. Filomat 31(2), 363—-370 (2017)

[64] Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M., Roselli, V.: Upward planar morphs. Algorithmica 82(10), 2985–3017 (2020)

[65] Davidson, R., Harel, D.: Drawing graphs nicely using simulated annealing. ACM Trans. Graph. 15(4), 301–331 (1996), https://doi.org/10.1145/234535.234538

[66] Demel, A., Dürrschnabel, D., Mchedlidze, T., Radermacher, M., Wulf, L.: A greedy heuristic for crossing-angle maximization. In: Biedl, T.C., Kerren, A. (eds.) Graph Drawing and Network Visualization. LNCS, vol. 11282, pp. 286–299. Springer (2018), https://doi.org/10.1007/978-3-030-04414-5_20

[67] Devanny, W.E., Kindermann, P., Löffler, M., Rutter, I.: Graph drawing contest report. In: Biedl, T.C., Kerren, A. (eds.) Graph Drawing and Network Visualization. vol. 11282, pp. 609–617. Springer (2018), https://doi.org/10.1007/978-3-030-04414-5_43

[68] Devkota, S., Ahmed, A.R., Luca, F.D., Isaacs, K.E., Kobourov, S.G.: Stress-Plus-X (SPX) graph layout. In: Archambault, D., Tóth, C.D. (eds.) Graph Drawing and Network Visualization. LNCS, vol. 11904, pp. 291–304. Springer (2019), https://doi.org/10.1007/978-3-030-35802-0_23

[69] Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing. Prentice Hall, Upper Saddle River, NJ (1999)

[70] Di Battista, G., Didimo, W.: Gdtoolkit. In: Tamassia, R. (ed.) Handbook on Graph Drawing and Visualization., pp. 571–597. Chapman and Hall/CRC, Boca Raton, FL, USA (2013)

[71] Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice-Hall (1999)

[72] Di Giacomo, E., Didimo, W., Liotta, G., Meijer, H.: Area, curve complexity, and crossing resolution of non-planar graph drawings. Theory Comput. Syst. 49(3), 565–575 (2011)

[73] Didimo, W.: Density of straight-line 1-planar graph drawings. Inf. Process. Lett. 113(7), 236–240 (2013)

[74] Didimo, W., Eades, P., Liotta, G.: Drawing graphs with right angle crossings. In: Dehne, F.K.H.A., Gavrilova, M.L., Sack, J., Tóth, C.D. (eds.) Workshop on Algorithms and Data Structures. LNCS, vol. 5664, pp. 206–217. Springer (2009)

[75] Didimo, W., Eades, P., Liotta, G.: A characterization of complete bipartite RAC graphs. Inf. Process. Lett. 110(16), 687–691 (2010)

[76] Didimo, W., Eades, P., Liotta, G.: Drawing graphs with right angle crossings. Theor. Comput. Sci. 412(39), 5156–5166 (2011)

[77] Didimo, W., Liotta, G., Montecchiani, F.: A survey on graph drawing beyond planarity. ACM Comput. Surv. 52(1), 4:1–4:37 (2019)

[78] Dujmovic, V., Gudmundsson, J., Morin, P., Wolle, T.: Notes on large angle crossing graphs. Chic. J. Theor. Comput. Sci. 2011 (2011), http://cjtcs.cs.uchicago.edu/articles/CATS2010/4/contents.html

[79] Duncan, C.A., Kobourov, S.G.: Polar coordinate drawing of planar graphs with good angular resolution. J. Graph Algorithms Appl. 7(4), 311–333 (2003), https://doi.org/10.7155/jgaa.00073

[80] Durocher, S., Gethner, E., Mondal, D.: Thickness and colorability of geometric graphs. Computational Geometry 56, 1–18 (2016), https://www.sciencedirect.com/science/article/pii/S092577211630013X

[81] Eades, P.: A heuristic for graph drawing. Congressus Numerantium 42, 149–160 (1984)

[82] Eades, P., Hong, S., Katoh, N., Liotta, G., Schweitzer, P., Suzuki, Y.: A linear time algorithm for testing maximal 1-planarity of graphs with a rotation system. Theor. Comput. Sci. 513, 65–76 (2013)

[83] Eades, P., Symvonis, A., Whitesides, S.: Three-dimensional orthogonal graph drawing algorithms. Discret. Appl. Math. 103(1-3), 55–87 (2000)

[84] Eades, P., Wormald, N.C.: Edge crossings in drawings of bipartite graphs. Algorithmica 11(4), 379–403 (1994), https://doi.org/10.1007/BF01187020

[85] Eppstein, D.: Twenty-one proofs of euler's formula. https://www.ics.uci.edu/~eppstein/junkyard/euler/, accessed: 2022-11-10

[86] Eppstein, D., van Kreveld, M.J., Mumford, E., Speckmann, B.: Edges and switches, tunnels and bridges. Comput. Geom. 42(8), 790–802 (2009)

[87] Erickson, J., Lin, P.: Planar and toroidal morphs made easier. In: Purchase, H.C., Rutter, I. (eds.) GD. LNCS, vol. 12868, pp. 123–137. Springer (2021), https://doi.org/10.1007/978-3-030-92931-2_9

[88] Euler, L.: Solutio problematis ad geometriam situs pertinentis. Commentarii Academiae Scientiarum Imperialis Petropolitanae 8, 128–140 (1736)

[89] Euler, L.: Elementa doctrinae solidorum. (1758)

[90] Fabrici, I., Harant, J., Madaras, T., Mohr, S., Soták, R., Zamfirescu, C.T.: Long cycles and spanning subgraphs of locally maximal 1-planar graphs. Journal of Graph Theory 95(1), 125–137 (2020)

[91] Floater, M., Gotsman, C.: How to morph tilings injectively. J. Comput. Appl. Math. 101, 117–129 (1999)

[92] Floater, M.S.: Parametrization and smooth approximation of surface triangulations. Comput. Aided Geom. Des. 14(3), 231–250 (1997)

[93] Formann, M., Hagerup, T., Haralambides, J., Kaufmann, M., Leighton, F.T., Symvonis, A., Welzl, E., Woeginger, G.J.: Drawing graphs in the plane with high resolution. In: 31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I. pp. 86–95. IEEE Computer Society (1990), https://doi.org/10.1109/FSCS.1990.89527

[94] Formann, M., Hagerup, T., Haralambides, J., Kaufmann, M., Leighton, F.T., Symvonis, A., Welzl, E., Woeginger, G.J.: Drawing graphs in the plane with high resolution. SIAM J. Comput. 22(5), 1035–1052 (1993), https://doi.org/10.1137/0222063

[95] Förster, H., Kaufmann, M.: On compact RAC drawings. In: Grandoni, F., Herman, G., Sanders, P. (eds.) European Symposium on Algorithms. LIPIcs, vol. 173, pp. 53:1–53:21. Schloss Dagstuhl (2020)

[96] Förster, H., Kaufmann, M., Raftopoulou, C.N.: Recognizing and embedding simple optimal 2-planar graphs. In: GD. Lecture Notes in Computer Science, vol. 12868, pp. 87–100. Springer (2021)

[97] de Fraysseix, H., Pach, J., Pollack, R.: How to draw a planar graph on a grid. Comb. 10(1), 41–51 (1990), https://doi.org/10.1007/BF02122694

[98] Frick, A., Ludwig, A., Mehldau, H.: A fast adaptive layout algorithm for undirected graphs. In: Tamassia, R., Tollis, I.G. (eds.) Graph Drawing. LNCS, vol. 894, pp. 388–403. Springer (1994), https://doi.org/10.1007/3-540-58950-3_393

[99] Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Softw., Pract. Exper. 21(11), 1129–1164 (1991)

[100] Fáry, I.: On straight lines representation of planar graphs. Acta Sci. Math. (Szeged) 11, 229–233 (1948)

[101] Gabow, H., Westermann, H.: Forests, frames, and games: Algorithms for matroid sums and applications. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. p. 407–421. STOC '88, Association for Computing Machinery, New York, NY, USA (1988), https://doi.org/10.1145/62212.62252

[102] Gansner, E.R., Koren, Y., North, S.C.: Graph drawing by stress majorization. In: Pach, J. (ed.) Graph Drawing. LNCS, vol. 3383, pp. 239–250. Springer (2004), https://doi.org/10.1007/978-3-540-31843-9_25

[103] Garcia-Moreno, E., Salazar, G.: Bounding the crossing number of a graph in terms of the crossing number of a minor with small maximum degree. Journal of Graph Theory 36(3), 168–173 (2001), https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-0118%28200103%2936%3A3%3C168%3A%3AAID-JGT1004%3E3.0.CO%3B2-%23

[104] Garey, M.R., Johnson, D.S.: Crossing number is np-complete. SIAM J. Algebraic Discrete Methods 4(3), 312–316 (sep 1983), https://doi.org/10.1137/0604033

[105] Germa, A., Heydemann, M., Sotteau, D.: Cycles in the cube-connected cycles graph. Discret. Appl. Math. 83(1-3), 135–155 (1998)

[106] van Goethem, A., Speckmann, B., Verbeek, K.: Optimal morphs of planar orthogonal drawings II. In: Archambault, D., Tóth, C.D. (eds.) Graph Drawing and Network Visualization, GD 2019. LNCS, vol. 11904, pp. 33–45. Springer (2019)

[107] Gotsman, C., Surazhsky, V.: Guaranteed intersection-free polygon morphing. Computers and Graphics 25, 67–75 (2001)

[108] Grigoriev, A., Bodlaender, H.L.: Algorithms for graphs embeddable with few crossings per edge. Algorithmica 49(1), 1–11 (2007)

[109] Grünbaum, B., Shepard, G.C.: The geometry of planar graphs. In: 8th British Combinatorial Conference (1981)

[110] Gutwenger, C., Mutzel, P.: Planar polyline drawings with good angular resolution. In: Whitesides, S. (ed.) Graph Drawing, 6th International Symposium, GD'98, Montréal, Canada, August 1998, Proceedings. LNCS, vol. 1547, pp. 167–182. Springer (1998), https://doi.org/10.1007/3-540-37623-2_13

[111] Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using NetworkX. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) Proceedings of the 7th Python in Science Conference. pp. 11 – 15 (2008)

[112] Heydari, A.A., Thompson, C.A., Mehmood, A.: Softadapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions. CoRR abs/1912.12355 (2019)

[113] Hlinený, P.: Crossing number is hard for cubic graphs. J. Comb. Theory, Ser. B 96(4), 455–471 (2006), https://doi.org/10.1016/j.jctb.2005.09.009

[114] Holyer, I.: The NP-completeness of edge-coloring. SIAM J. Comput. 10(4), 718–720 (1981)

[115] Hopcroft, J., Tarjan, R.E.: Efficient planarity testing. J. ACM 21(4) (1974)

[116] Hromkovic, J.: Design and Analysis of Randomized Algorithms - Introduction to Design Paradigms. Texts in Theoretical Computer Science. An EATCS Series, Springer (2005), https://doi.org/10.1007/3-540-27903-2

[117] (https://math.stackexchange.com/users/215955/moritz), M.: Graph theory: minors vs topological minors. Mathematics Stack Exchange (2022), uRL:https://math.stackexchange.com/q/1160040 (version: 2015-02-22)

[118] Huang, W., Eades, P., Hong, S.: Larger crossing angles make graphs easier to read. J. Vis. Lang. Comput. 25(4), 452–465 (2014)

[119] Huang, W., Eades, P., Hong, S., Lin, C.: Improving multiple aesthetics produces better graph drawings. J. Vis. Lang. Comput. 24(4), 262–272 (2013), https://doi.org/10.1016/j.jvlc.2011.12.002

[120] Huang, W., Hong, S., Eades, P.: Effects of crossing angles. In: PacificVis. pp. 41–46. IEEE Computer Society (2008), https://doi.org/10.1109/PACIFICVIS.2008.4475457

[121] Hudson, Alan, R.: Planar graphs : a historical perspective. (2004)

[122] Jordan, C.: Cours d'Analyse de l' Ecole Polytechnique, pp. 587–594. Gauthier-Villars (1887)

[123] Jünger, M., Mutzel, P.: 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. J. Graph Algorithms Appl. 1(1), 1–25 (1997), https://doi.org/10.7155/jgaa.00001

[124] Kainen, P.: Thickness and coarseness of graphs. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 39, 88–95 (sept 1973)

[125] Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Inf. Process. Lett. 31(1), 7–15 (1989), https://doi.org/10.1016/0020-0190(89)90102-6

[126] Kant, G.: Drawing planar graphs using the canonical ordering. Algorithmica 16(1), 4–32 (1996), https://doi.org/10.1007/BF02086606

[127] Katheder, J., Kobourov, S.G., Kuckuk, A., Pfister, M., Zink, J.: Simultaneous drawing of layered trees. In: 39th European Workshop on Computational Geometry (EuroCG'23) (2023)

[128] Kaufmann, M., Ueckerdt, T.: The density of fan-planar graphs. CoRR abs/1403.6184v1 (2014), http://arxiv.org/abs/1403.6184v1

[129] Kaufmann, M., Ueckerdt, T.: The density of fan-planar graphs. Electron. J. Comb. 29(1) (2022)

[130] Kaufmann, M., Wagner, D. (eds.): Drawing Graphs, Methods and Models, LNCS, vol. 2025. Springer (2001), https://doi.org/10.1007/3-540-44969-8

[131] Kawarabayashi, K.i., Reed, B.: Computing crossing number in linear time. In: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing. p. 382–390. STOC '07, Association for Computing Machinery, New York, NY, USA (2007), https://doi.org/10.1145/1250790.1250848

[132] Kindermann, P., Mchedlidze, T., Rutter, I.: Graph drawing contest report. In: Archambault, D., Tóth, C.D. (eds.) Graph Drawing and Network Visualization. vol. 11904, pp. 575–583. Springer (2019), https://doi.org/10.1007/978-3-030-35802-0_43

[133] Kleist, L., Klemz, B., Lubiw, A., Schlipf, L., Staals, F., Strash, D.: Convexity-increasing morphs of planar graphs. Comput. Geom. 84, 69–88 (2019)

[134] Knuth, D.E.: Computer-drawn flowcharts. Commun. ACM 6(9), 555–563 (1963)

[135] Korzhik, V.P.: Minimal non-1-planar graphs. Discrete Mathematics 308(7), 1319–1327 (2008), https://www.sciencedirect.com/science/article/pii/S0012365X07002087

[136] Korzhik, V.P., Mohar, B.: Minimal obstructions for 1-immersions and hardness of 1-planarity testing. J. Graph Theory 72(1), 30–71 (2013)

[137] Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29, 1–27 (1964), https://doi.org/10.1007/BF02289565

[138] Kuratowski, K.: Sur le problème des courbes gauches en topologie. Fund. Math. 15, 217–283 (1930)

[139] Lazard, S., Lenhart, W.J., Liotta, G.: On the edge-length ratio of outerplanar graphs. Theor. Comput. Sci. 770, 88–94 (2019), https://doi.org/10.1016/j.tcs.2018.10.002

[140] Leiserson, C.E.: Area-efficient graph layouts. In: 21st Annual Symposium on Foundations of Computer Science (sfcs 1980). pp. 270–281 (1980)

[141] Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is np-complete. Journal of Computer and System Sciences 20(2), 219–230 (1980), https://www.sciencedirect.com/science/article/pii/0022000080900604

[142] Lin, C., Yen, H.: A new force-directed graph drawing method based on edge-edge repulsion. J. Vis. Lang. Comput. 23(1), 29–42 (2012), https://doi.org/10.1016/j.jvlc.2011.12.001

[143] Liotta, G., Montecchiani, F.: L-visibility drawings of IC-planar graphs. Inf. Process. Lett. 116(3), 217–222 (2016)

[144] Malitz, S.M., Papakostas, A.: On the angular resolution of planar graphs. In: Kosaraju, S.R., Fellows, M., Wigderson, A., Ellis, J.A. (eds.) Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada. pp. 527–538. ACM (1992), https://doi.org/10.1145/129712.129764

[145] Mansfield, A.: Determining the thickness of graphs is NP-hard. Mathematical Proceedings of the Cambridge Philosophical Society 93(1), 9–23 (1983)

[146] Mendelson, B.: Introduction to Topology: Third Edition. Dover Books on Mathematics, Dover Publications (2012), https://books.google.de/books?id=FWFmoEUJSwkC

[147] Motwani, R., Raghavan, P.: Randomized algorithms. In: Atallah, M.J. (ed.) Algorithms and Theory of Computation Handbook. Chapman & Hall/CRC Applied Algorithms and Data Structures series, CRC Press (1999), https://doi.org/10.1201/9781420049503-c16

[148] Mutzel, P.: An alternative method to crossing minimization on hierarchical graphs. In: North, S. (ed.) Graph Drawing. pp. 318–333. Springer Berlin Heidelberg, Berlin, Heidelberg (1997)

[149] Nash-Williams, C.S.A.: Edge-disjoint spanning trees of finite graphs. Journal of the London Mathematical Society s1-36(1), 445–450 (jan 1961)

[150] Pach, J., Tóth, G.: Graphs drawn with few crossings per edge. Comb. 17(3), 427–439 (1997), https://doi.org/10.1007/BF01215922

[151] Purchase, H.C.: Which aesthetic has the greatest effect on human understanding? In: Di Battista, G. (ed.) Graph Drawing, 5th International Symposium, GD '97, Rome, Italy, September 18-20, 1997, Proceedings. LNCS, vol. 1353, pp. 248–261. Springer (1997), https://doi.org/10.1007/3-540-63938-1_67

[152] Purchase, H.C.: Effective information visualisation: a study of graph drawing aesthetics and algorithms. Interact. Comput. 13(2), 147–162 (2000)

[153] Purchase, H.C.: Metrics for graph drawing aesthetics. J. Vis. Lang. Comput. 13(5), 501–516 (2002), https://doi.org/10.1006/jvlc.2002.0232

[154] Radermacher, M., Reichard, K., Rutter, I., Wagner, D.: A geometric heuristic for rectilinear crossing minimization. In: Pagh, R., Venkatasubramanian, S. (eds.) ALENEX. pp. 129–138. SIAM (2018), https://doi.org/10.1137/1.9781611975055.12

[155] Rahmati, Z., Emami, F.: RAC drawings in subcubic area. Inf. Process. Lett. 159-160, 105945 (2020)

[156] Richter, R.B., Širáň, J.: The crossing number of k3, n in a surface. Journal of Graph Theory 21(1), 51–54 (1996), https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0118%28199601%2921%3A1%3C51%3A%3AAID-JGT7%3E3.0.CO%3B2-L

[157] Robertson, N., Seymour, P.D.: Graph minors. xx. wagner's conjecture. Journal of Combinatorial Theory, Series B 92, 325–357 (2004)

[158] Robertson, N., Seymour, P.: Graph minors .xiii. the disjoint paths problem. Journal of Combinatorial Theory, Series B 63(1), 65–110 (1995), https://www.sciencedirect.com/science/article/pii/S0095895685710064

[159] Robinson, D.F.: The description of rectanguloid curves. In: 8th British Combinatorial Conference (1981)

[160] Schaefer, M.: The graph crossing number and its variants: A survey. Electronic Journal of Combinatorics 1000 (2013)

[161] Schaefer, M.: Rac-drawability is ∃ R-Complete. In: Purchase, H.C., Rutter, I. (eds.) Graph Drawing and Network Visualization. LNCS, vol. 12868, pp. 72–86. Springer (2021)

[162] Schneck, T.: New Parameters for Beyond-Planar Graphs. dissertation, University Tübingen (2020), https://publikationen.uni-tuebingen.de/xmlui/handle/10900/109680

[163] Schnyder, W.: Embedding planar graphs on the grid. In: Proceedings of the 1st ACM-SIAM Symposium on Discrete Algorithms (SODA '90). pp. 138–148 (1990)

[164] Shepard, R.N.: The analysis of proximities: Multidimensional scaling with an unknown distance function. I. Psychometrika 27, 125–140 (1962), https://doi.org/10.1007/BF02289630

[165] Sugiyama, K., Tagawa, S., Toda, M.: Methods for visual understanding of hierarchical system structures. IEEE Trans. Syst. Man Cybern. 11(2), 109–125 (1981), https://doi.org/10.1109/TSMC.1981.4308636

[166] Suzuki, Y.: Optimal 1-planar graphs which triangulate other surfaces. Discrete Math. 310(1), 6–11 (2010)

[167] Sýkora, O., Vrto, I.: On crossing numbers of hypercubes and cube connected cycles. BIT 33(2), 232–237 (1993)

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

[169] Tamassia, R. (ed.): Handbook on Graph Drawing and Visualization. Chapman and Hall/CRC (2013), https://www.crcpress.com/Handbook-of-Graph-Drawing-and-Visualization/Tamassia/9781584884125

[170] Therese Biedl, Markus Chimani, M.D., Mutzel, P.: Crossing number for graphs with bounded pathwidth. Algorithmica 82(2), 355–384 (2020)

[171] Thomassen, C.: Deformations of plane graphs. J. Comb. Theory Ser. B 34(3), 244–257 (1983)

[172] Thomassen, C.: Rectilinear drawings of graphs. Journal of Graph Theory 12(3), 335–341 (1988), https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.3190120306

[173] Thomassen, C.: The graph genus problem is np-complete. Journal of Algorithms 10(4), 568–576 (1989), https://www.sciencedirect.com/science/article/pii/0196677489900060

[174] Tóth, C.D.: On rac drawings of graphs with two bends per edge. arXiv preprint arXiv:2308.02663 (accepted at GD2023, to be published) (2023)

[175] Tutte, W.T.: How to draw a graph. Proceedings of the London Mathematical Society s3-13(1), 743–767 (1963)

[176] Ugander, J., Backstrom, L.: Balanced label propagation for partitioning massive graphs. In: Leonardi, S., Panconesi, A., Ferragina, P., Gionis, A. (eds.) WSDM. pp. 507–516. ACM (2013), https://doi.org/10.1145/2433396.2433461

[177] u/noktulo: Orthogonal new york city subway map (2019), https://www.reddit.com/r/MapPorn/comments/az8vcd/orthogonal_new_york_city_subway_map/, [Online; accessed 21.04.2023]

[178] Urschel, J.C., Wellens, J.: Testing gap k-planarity is np-complete. Inf. Process. Lett. 169, 106083 (2021)

[179] Wagner, K.: Über eine eigenschaft der ebenen komplexe. Math. Ann. 114, 570–590 (1937)

[180] Wagner, K.: Bemerkungen zum Vierfarbenproblem. Jahresbericht der Deutschen Mathematiker-Vereinigung 46, 26–32 (1936)

[181] Ware, C., Purchase, H.C., Colpoys, L., McGill, M.: Cognitive measurements of graph aesthetics. Inf. Vis. 1(2), 103–110 (2002), https://doi.org/10.1057/palgrave.ivs.9500013

[182] Whitney, H.: Congruent graphs and the connectivity of graphs. Hassler Whitney Collected Papers pp. 61–79 (1992)

[183] Yannakakis, M.: Node-and edge-deletion np-complete problems. In: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing. p. 253–264. STOC '78, Association for Computing Machinery, New York, NY, USA (1978), https://doi.org/10.1145/800133.804355

[184] Ziegler, G.M.: Steinitz' Theorem for 3-Polytopes, pp. 103–126. Springer New York, New York, NY (1995), https://doi.org/10.1007/978-1-4613-8431-1_4

Appendix A

Other works of the author

The following contains a list of published collaborative work of the author which was not covered in the thesis:

- In [10], we study Turán-Type questions for bipartite graphs in the field of beyond-planarity. Namely, we establish upper bounds on the edge-density of bipartite IC-, NIC-, RAC-, fan-planar and 2-planar graphs. We also develop a Crossing Lemma (restricted to bipartite graphs) which, with the help of our result for 2-planar graphs, establishes that n-vertex k-planar graphs can have at most $3.005\sqrt{kn}$ edges, whereas the previous best upper bound was $3.81\sqrt{kn}$ edges [1].

- In [58], we considered the following question. Given a geodetic graph G, i.e., a graph where the shortest path between any two vertices is unique, is there a so called *philogeodetic* drawing of G, i.e., a drawing where any two shortest paths *meet* at most once. A meet is either a crossing in the drawing or a maximal interval of vertices/edges that the shortest path share. We answer this question in the negative by providing an explicit construction (a subdivision of a complete graph) with the use of a charging scheme on the number of crossings. In fact, by subdividing every edge sufficiently many times, the counter-example has an edge-vertex-ratio of only $1 + \epsilon$. On the other hand, we show a diameter-2 counterexample to the question. Finally, by combining edge-vertex-ratio and diameter, we show that all geodetic diameter-2 graphs with edge-vertex-ratio less than 1.5 admit a philogeodetic drawing.

- In [127], we consider the crossing-minimization problem for the simultaneous embedding of layered trees. The problem is known to be NP-hard for linearly many trees on two layers. Let n be the total number of vertices of all trees. We provide an $\mathcal{O}(n^2)$ algorithm based on dynamic programming for two trees on arbitrary many layers and an $\mathcal{O}(n^{\mathcal{O}(k)})$ algorithm for k-trees on 3-layers.